A Collaborative Mobile Sensing Framework using Mobile Cloud Computing

Radhika Loomba, B.Tech (Hons.)

Department of Computing and Mathematics,
School of Science and Computing

Waterford Institute of Technology

This dissertation is submitted for the degree of

Doctor of Philosophy

Supervisors: Dr Lei Shi and Dr Brendan Jennings

May 2017

‘To strive, to seek, to find, and not to yield.’

I dedicate this dissertation to my parents and my younger sister, who inspired me and gave me the courage to follow my dreams.

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save to the extent that such work has been cited and acknowledged within the text of my work.

Radhika

Radhika Loomba, B.Tech (Hons.)
Student ID: 20057666
May 2017

Acknowledgements

This dissertation is the result of the prayers, love, and guidance of people from around the world who never-failed to believe in me, and even when I faltered, relentlessly devoted their time and efforts to keep me motivated. I take this opportunity to express my heart-felt gratitude and thank all of them for always being there for me.

I begin by thanking my supervisors, Dr. Lei Shi and Dr. Brendan Jennings for their invaluable expertise, guidance, and patience throughout these years. It is by their example that I have learnt to persevere through failure and to devote myself untiringly towards creating impactful research. I am also thankful to the entire team at the Telecommunications Software & Systems Group, and the wider community at the Waterford Institute of Technology, especially Prof. Willie Donnelly, who have been instrumental in instilling a commitment towards excellence in research, innovation, and personal growth.

I also extend my gratitude to my enterprise mentors at Intel Labs, Europe and the Irish Research Council for funding my study and giving me the opportunity to learn from both academia and industry.

I thank my parents, Dr. Kapil Loomba and Dr. Anju Loomba, my sister Gayatri, my grandparents, and my entire extended family for being a constant source of moral and emotional support. Their endless words of encouragement, at all times of the day and night, gave me the wings to pursue my dreams. Last but not the least, I also thank all my friends from India, Ireland, and from around the world for their faith in me and for the endless phone calls/emails to keep me inspired and motivated.

I couldn’t have done it without you all.

Abstract

The ubiquity of mobile devices, tablets, wearables and other smart devices, with ever-more capable embedded sensors, has accelerated the use of mobile sensing applications. These applications harness the processed sensor data to offer a more context-aware and personalized user-experience for purposes including healthcare, entertainment, environmental monitoring and promoting socialization.

However, the resource limitation of mobile devices creates a significant bottleneck for the realization of such applications. In this scenario, Mobile Cloud Computing (MCC) is a promising solution that allows the offload of raw or pre-processed sensed data to the application logic hosted in the cloud. This research utilizes the scalability and processing capabilities of MCC to present an intelligent framework that readily supports collaborative sensing and provides efficient collection of the sensed data, whilst conserving the energy of the mobile devices and meeting the performance constraints of applications.

The benefits of this collaborative sensing framework include the capability to satisfy multiple mobile applications with optimal sensing using minimal number of mobile devices, context-driven reporting of sensed data to the application cloud, and achieving all of this in an energy efficient manner.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Publications

- Loomba, R., Shi, L., Jennings, B., Friedman, R., Kennedy, J., & Butler, J. Information Aggregation for Collaborative Sensing in Mobile Cloud Computing. In *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 149–158, April 2014. (Loomba et al., 2014b)
- Loomba, R., Shi, L., & Jennings, B. State-machine driven opportunistic sensing by mobile devices. In *IEEE Global Communications Conference (GLOBECOM)*, pp. 2739—2744, December 2014. (Loomba et al., 2014a)
- Loomba, R., de Fréin, R., & Jennings, B. Selecting Energy Efficient Cluster-Head Trajectories for Collaborative Mobile Sensing. In *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, December 2015. (Loomba et al., 2015a)
- Loomba, R., Shi, L., Jennings, B., Friedman, R., Kennedy, J., & Butler, J. Energy-aware collaborative sensing for multiple applications in mobile cloud computing. *Sustainable Computing: Informatics and Systems*, 8, pp. 47–59, 2015. (Loomba et al., 2015b)
- Loomba, R., Shi, L., & Jennings, B. State-machine driven Collaborative Mobile Sensing Serving Multiple Internet-of-Things Applications. In *15th IFIP/IEEE International Symposium on Integrated Network Management (IM 2017) -Special Track on Management of IoT*, to appear, 2017. (Loomba et al., 2017)

Chapter 1

Introduction

The world is increasingly focused on always being connected, leading to the proliferation of smartphones, tablets, wearables and other smart devices. Such innovation accelerates the ability to exploit mobile devices as personal digital assistants and they rapidly become the most effective means of communication for all human-to-human, human-to-machine and even machine-to-machine interactions. Furthermore, the embedded sensor systems in mobile devices provide an increasingly rich set of sensors, covering environmental sensing (barometers, photometers, and thermometers), motional sensing (accelerometers, gravity sensors and gyroscopes) and positional sensing (compasses and magnetometers). Besides encouraging such sensor development, mobile technology has enabled new data-gathering capabilities providing the much-desired connectivity for sensors present practically anywhere. Examining global mobile data trends, Cisco highlights how mobile data traffic will increase eightfold between 2015 and 2020 (Cisco), given the ubiquity of mobile devices. This wide-spread availability sets the stage for mobile sensing and mobile sensing applications to harness sensor data for influencing daily life events. Currently, two contrasting sensing paradigms, participatory and opportunistic sensing, have been explored (Lane et al., 2008); each with their own benefits and shortcomings that affect application deployment and diversity. They are remarkably scalable and affordable, given the wide proliferation of cellular phone infrastructure and straightforward programmability. Multiple mobile applications use the capability of these paradigms to offer a personalized experience by sending context feedback to mobile users and a global trend of increased time spent on mobile applications has emerged. These applications become part of the every-day lifestyle of the user by either tracking environmental factors like temperature, humidity, pressure and the presence of unwanted gases, or reacting to rising noise levels for traffic management or promoting social interactions. However, the resource limitations of mobile

devices creates the biggest bottleneck in such scenarios due to the computational capabilities required by the applications.
'Mobile Cloud Computing'(MCC) attempts to create a new paradigm for combating this issue. In its broadest sense, MCC refers to the application of cloud computing technology to the hosting and execution of mobile applications. MCC proposals range from the provision of computation and storage resources as a 'data center in a box' at a wireless access router (Satyanarayanan et al., 2009) to the creation of a 'fog' of virtualized computation, storage and networking resources provided by large numbers of network edge devices (Shi et al., 2012a). From a resource management perspective, MCC can be viewed as an extension of the traditional data center centric scope of cloud computing to encompass the provision of computational and storage resources by devices at the network edge. The powerful cloud servers are utilized for offloading storage and data processing operations, enabling better performance. As mobile applications continue to become more computation intensive, such offloading seems to present a viable solution. They assist in reducing the computation (Rudenko et al., 1998) involved in mobile sensing thereby improving energy-efficiency (Kumar and Lu, 2010).
This convergence of mobile technology, data analytics and mobile cloud computing sets the stage for dramatically impacting many sectors such as healthcare, social networks, traffic estimation and environmental monitoring (Khan et al., 2013; Lane et al., 2010). In this setting, an important technical challenge is the establishment of a collaborative mobile sensing framework which utilizes the scalability and processing capabilities of the mobile cloud and provides efficient collection of the sensed data in terms of energy efficiency and monitoring accuracy. This dissertation defines the scope using the research work that has been done to quantify the envisaged framework and presents multiple algorithms that cover both urban and sub-urban scenarios.
The remainder of this chapter presents the formal hypothesis that summates the intent of this dissertation, explicitly defines scope using research questions, states the research contributions and outlines the organization of this dissertation.

1.1 Research Hypothesis

With the ubiquity and heterogeneity of mobile devices interacting with IoT sensor devices, it becomes infeasible to follow the approach of connecting applications directly to the devices. This is mainly because it is unscalable to maintain one-to-one relations, when multiple applications are requesting for sensed data from an individual device that needs to

concurrently collect, pre-process and offload to satisfy all requests. For such a scenario, this research focuses on creating a common platform that is scalable, supports interoperability and meets the multiple performance and security constraints of various applications. It should also enable efficient sensor data collection, analysis and dissemination from the energy-constrained mobile devices.
Formally, the research hypothesis is presented as:

“A collaborative mobile sensing framework utilizing mobile cloud computing technology will help in improving energy-efficiency of mobile devices while providing accurate context-aware, real-time sensed data information to several applications."

This collaborative sensing framework can be viewed to consist of various distinct but complimentary components, where each component must work in synchronization for better performance and energy utilization of the mobile devices. This leads to dedicated components that optimally map application requirements to the capabilities of the embedded sensors in mobile devices, calculate the minimal number of mobile devices needed for satisfying all requirements, efficiently analyse context for improved accurate reporting of sensed data in real-time as well as intelligently select one or more mobile devices to offload the collected data to the mobile cloud for further processing. Thus, some features of the collaborative framework include:

- Removing resource and processing limitations of mobile devices by using the mobile cloud.
- Introducing a trustworthy framework that interacts with sensors on the mobile device for delivering an enhanced environment for mobile applications.
- Implementing algorithms that reduce data redundancy via mobile offloading techniques.
- Collecting context-aware information from mobile devices to assist applications.
- Efficiently localizing contextual changes to support appropriate mitigation actions when necessary.
- Replacing continuous sensing systems with an intelligent system for energy saving.
- Adapting clustering algorithms for collaboration between mobile devices.

- Optimizing selection of suitably positioned cluster-heads to offload aggregated sensed data to the application cloud server.
- Interacting with available traditional sensors and wearable technology in the surrounding environment for better collection and processing of sensed data.

Five research questions have been examined in this dissertation to ensure these features. The intent of this work is to benefit from addressing each question to iteratively improve the collaborative framework and the involved subsequent research.

1.1.1 First Research Question (RQ1)

How should the collaborative framework be modelled to facilitate seamless interaction between multiple applications and the mobile devices located within a particular physical area?
Since the need of a collaborative framework is apparent, the first research question addresses the decisions that have to be taken regarding the architectural design and placement of the framework. Such a framework will act like an abstract layer and involves several inherent aspects that need to be resolved. These include, but are not limited to, issues pertaining to identification and localization of mobile devices and their embedded sensors, management of network parameters like bandwidth/throughput when communicating between applications, mobile devices and the cloud servers as well as decisions relating to the placement of logic to handle incoming and outgoing data flows.
The framework must also be able to support autonomous task allocation and scheduling for these applications by identifying which physical areas have to be covered and what mobile devices are in a position to capture the required sensor data. Furthermore, the feature of mobile sensing that distinguishes it from the traditional sensor networks is the absence of single data ownership, introducing concerns regarding content integrity and security. Thus, it is imperative that the sensing framework is able to support privacy mechanisms and send authentic sensor data, when required, without revealing sensitive information stored on the mobile devices regarding the identities of the users. This leads to the requirement of a generic framework that is able to adapt and scale with different scenarios.

1.1.2 Second Research Question (RQ2):

In such a framework, how many of these devices are sufficient to sense the data without introducing unnecessary redundancy?

Today's digital landscape is predominantly driven by various mobile applications. Propelled by the growing ubiquity of diverse sensors embedded into mobile devices, these applications have the potential to affect several sectors such as industrial manufacturing, environmental monitoring, healthcare, sport equipments and training, transport and logistics, tourism, and social network services.

With this increase in the number of mobile applications, there exists a high probability that several applications working in the same environment would request for similar data from the sensors. However, acquisition, aggregation, processing and storage, of such highly co-related sensor data by mobile devices, have huge costs in terms of energy expended during these processes. This may result in faster energy depletion, network overloading, latency issues and even congestion due to the increased traffic of a large numbers of mobile devices offloading data to the cloud. In this scenario, the second research question is aimed at determining an adequate number of mobile devices that must be activated by the framework to ensure optimal energy usage whilst maintaining the accuracy required by the applications.

1.1.3 Third Research Question (RQ3):

How can application requirements be encoded to determine when a mobile device should sense and offload sensed data whilst saving its energy?

The advantages of multiple mobile sensors providing context feedback to support personalization of mobile applications has been studied extensively. This has inspired advancements in mobile handset technology to include an ever increasing incorporation of sophisticated sensors such as cameras, accelerometers, gyroscopes and compasses along with capable programmable interfaces to allow interaction of applications directly with these sensors. From the application perspective, this has the added benefit of providing a source of context information without requiring additional investment or deployment of dedicated sensor devices.

Thus, this research question focuses on three key challenges which are identifying the contexts for which data is required by the applications, localising the contextual changes in the environment and minimising the energy expended in reporting sensed data. The former two are important since most mobile applications are event-driven and appropriate mitigation actions can be taken, when necessary, by localizing the physical area affected. The latter is a particular problem when data is collected and reported at a fixed rate, namely continuous sensing. This is significant as the standby time of a device can be reduced from twenty hours to six hours upon deployment of applications using continuous sensing strategies (Miluzzo et al., 2008). Furthermore, this introduces the challenges for security, ensuring consistency in sensor data, detection of context for targeted sampling of data, and sensor mobility. Depending on the architecture, it is important for either the sensing device or the middleware to understand when information can be captured and what accuracy can be delivered.

1.1.4 Fourth Research Question (RQ4):

How and when should sensed information from surrounding IoT sensors and wearables be collected by the framework?
The Internet of Things paradigm shows the potential to rapidly transform various spheres of human life. It envisages a vast web of connected sensor systems that analyse, communicate and coordinate collected data to improve performance. This includes not only commercial computational devices, but also multiple interoperable, small, lower power and less expensive devices surrounding a mobile user. This means that mobile applications are in a position to profit from this wide-spread heterogeneity to collect sensor information from otherwise-missing sensors and even acquire data in real-time with higher accuracy. However, the added cost of communicating with the IoT sensor must be considered. Thus, this research question addresses the trade-off between the number and quality of sensors accessible by the applications and the energy-efficiency of the mobile devices. The framework needs to intelligently decide how and when the surrounding sensors are accessed in a secure manner.

1.1.5 Fifth Research Question (RQ5):

How can the framework be adapted to work in a communication limited scenario whilst offloading accurate sensed data for applications?
Mobile devices contain radio interfaces for several communications technologies, including short-range transmission mediums like BlueTooth and long-range transmission mediums like WiFi, as well as cellular technology. Each of these differ in availability, coverage range and performance. Additionally, depending on the communication medium, mobile devices might expend more energy.
However, depending on the scenario, the usage of these technologies might be limited and there must be an alternative of performing the required actions in any communication medium. This is also imperative for proper energy management and energy consumed for transmission of data via different communication mediums is vastly different. This research question highlights the need to address collaborative techniques between devices in such

circumstances. It is focused towards investigating mechanisms that can adapt the sensing framework, present accurate sensor data and remain energy-efficient for all mobile devices.

1.2 Research Contribution

The main contribution of this dissertation is the design and execution of the collaborative sensing framework that uses mobile cloud technology to seamlessly interact and assuage the different requirements of multiple mobile applications with varying demands related to physical sensing areas covered, sensing and reporting rates, and the availability of communication mediums. Each of the research questions presented above are instrumental in building this framework and have been addressed in detail in Chapter 3, Chapter 4 and Chapter 5.

This section extracts and summarises the high level contributions that have been made in this dissertation.

1. A centralized middleware architecture is defined for the framework that mediates between the multiple applications requesting for sensed data and the mobile devices present in a particular physical area. Such an architecture is crucial as no coordination is needed among the mobile device users which is a desirable quality for the collaborative framework.

2. The best combination of sensors to activate on the available devices, taking into account those devices' predicted locations, energy status and the requirements of multiple applications is identified in Algorithm Info-Aggregation (Algorithm 3.2). It is defined for a pico-cell deployment for the collaborative sensing framework and uses a revised version of the frequent pattern mining technique to efficiently calculate the minimal set of involved mobile devices that cover application constraints, thereby reducing redundancy in offloaded sensed data streams. Using this methodology, it is shown that the number of devices offloading sensed data in the cloud and the volume of the offloaded data can be reduced whilst also saving energy consumed by the mobile devices.

3. The capabilities of the framework are extended in Algorithms Context-Localize (Algorithm 4.1) and Assisted-Aggregation (Algorithm 4.2), by adding knowledge regarding the event-driven nature of most mobile applications. The concept of application-specific state machines is introduced to quantify the context surrounding a

mobile device user and this information is leveraged for efficiently offloading sensed data only when particular events occur. Both of the mentioned algorithms successfully detect and automatically deliver sensed information to concerned applications or systems in an energy-efficient manner. Algorithm Context-Localize is presented for an environmental monitoring application in an enterprise environment where mobile devices are automatically clustered based on their location, with a cluster-head reporting sensed data to the application logic. Algorithm Assisted-Aggregation covers the case when multiple applications request for sensed data with varying contexts in an urban city scenario and uses consolidates state machines, maintained for small physical areas, to cover all reporting constraints. It harnesses the advantages of frequent pattern mining combined with these state machines to reduce the number of active mobile devices and volume of the offloaded data.

4. Algorithm Assisted-Aggregation (Algorithm 4.2) further shows how a mobile device can act as a travelling agent and collect sensed data from surrounding sensors in an IoT-inspired scenario to create a context-rich environment. This algorithm intelligently determines when the capabilities of the embedded sensors are limited in providing the needed accuracy for an application. As the mobile user move within different sensing areas packed with heterogeneous sensors and numerous wireless radios, seamless communication is attempted with these surrounding devices to estimate the trade-off between the requested accuracy and energy expended in collecting data from the external device.

5. For scenarios with unavailable WiFi, two algorithms namely, Algorithm CH-Trajectory (Algorithm 5.1) and Algorithm CH-Trajectory-With-Forgetting-Factor (Algorithm 5.2) are presented that stochastically select cluster-heads by exploiting knowledge of the communication distance between the devices and the cellular base stations, to reduce the number of devices offloading over longer distances for energy efficiency. By treating the selection of the cluster-head as a separate problem from the data compression, these algorithms minimize the trajectory of a cluster-head successively and allow knowledge of good cluster-head decisions to feed-forward into future selection decisions. This provides the framework with the advantage of providing real-time access to sensor data with energy efficiency even for areas with restricted communication medium access.

Table 1.1 Research Contributions

Research Question	Research Contribution	Chapter and Section
RQ1	Centralized Architecture for Collaborative Sensing Framework	Chapter 3, §3.2
RQ2	Algorithm Info-Aggregation	Chapter 3, §3.4.2
RQ3	Algorithm Context-Localize and Algorithm Assisted-Aggregation	Chapter 4, §4.2 & §4.3.2
RQ4	Algorithm Assisted-Aggregation	Chapter 4, §4.3.2
RQ5	Algorithm CH-Trajectory and Algorithm CH-Trajectory-With-Forgetting-Factor	Chapter 5, §5.3

1.3 Dissertation Organization

This document is structured as follows. Chapter 2 first presents background information in the core fields of mobile sensing and mobile cloud computing while highlighting the challenges of both fields. Next, a literature review is presented that sequentially describes the related work done with respect to the areas of contribution in this dissertation, namely collaborative models, aggregation techniques, context-awareness with energy efficiency, mobility models and cluster-head selection algorithms.

The following three chapters address the research questions in detail. Table 1.1 maps the research questions with the equivalent contribution and also indicates the chapter and section in which it is explained.

Chapter 3 addresses the first research question (RQ1) and the second research question (RQ2). It presents a centralized architecture for the sensing framework to support sharing of resources between co-located mobile devices, and identifies the potential of multi-tasking the capabilities of a mobile device. It defines and evaluates an algorithm that studies the trade-off between accuracy of application requirements, energy expended by the devices and the volume of offloaded data.

Chapter 4 addresses the third research question (RQ3) and the fourth research question (RQ4). It introduces the knowledge of event-driven mobile applications into the framework with the help of application-specific state machines. It defines and evaluates two algorithms that use this information to reduce energy expended during sensing and reporting of the sensed data to multiple applications. It also presents the mobile device as a gateway for IoT scenarios, that can communicate with surrounding devices.

Chapter 5 addresses the fifth research question (RQ5) and extends the framework by presenting a clustering approach for devices to select cluster-heads to offload sensed data over longer transmission distances using the cellular radio. It defines and evaluates two algorithms that use the knowledge regarding transmission distance to a cellular base station to modulate the trajectory of the cluster-head by selecting suitably positioned devices as cluster-heads.

Finally, Chapter 6 concludes this dissertation by summarizing the chapters and presenting future work.

Chapter 2

Background and Related Work

This chapter provides an in-depth study on the technology and trends that are relevant for effectively building the collaborative sensing framework.

The first section of the chapter explores the advancements and challenges of mobile sensing in §2.1.1 and mobile cloud computing in §2.1.2 that have inspired this research work. Both these fields are fast-paced, self-accelerating technologies and although the state-of-art is continuously evolving, this study helps in identifying knowledge gaps that demand further investigation. Since mobility is an inherent characteristic of mobile devices on which this research is based, the framework also needs to embody mobility models. Thus, this background study also presents available mobility models that can help in simulating human movement in §2.1.3.

The next section of this chapter is a literature review of the areas of direct relevance to the features of the collaborative sensing framework. It begins by describing the work that has been done on collaborative models in §2.2.1 and presents existing methodologies in the domain of aggregation in §2.2.2. Subsequent subsections focus on context-awareness with energy-efficient algorithms in §2.2.3 and efficient cluster-head selection techniques in §2.2.4. The chapter concludes with a summary in §2.3 that compares and contrasts between the present literature and the proposed work of this dissertation.

2.1 Background

This section describes research in the core areas of mobile sensing and mobile cloud computing, that form the basis of this dissertation. Additionally, mobility models are also presented since the movement of mobile devices needs to be simulated for this research.

2.1.1 Mobile Sensing

A sensor can be defined as a converter that measures a physical quantity and converts it into a signal, which can be read by an observer or an instrument. When sensors first got incorporated within mobile devices, the components were merely tools to facilitate interaction with the device. However, influenced by the proliferation of mobile devices and the availability of cheap embedded sensors, mobile phones are becoming highly assorted collections that collect, process and disseminate sensed data (Lane et al., 2010; Zhang et al., 2016b). This was further accelerated by the adoption and availability of smaller and faster multi-chip core technologies to process data simultaneously and more accurately. These sensors include specialized environmental sensors (ambient light, barometers, photometers, and thermometers), motion sensors (accelerometers, gravity sensors and gyroscopes), positional sensors (compasses and magnetometers) as well as general purpose sensors such as microphones, proximity sensors and cameras. Even more sophisticated sensors such as gas sensors and humidity sensors are soon to be incorporated by manufacturing companies like Sensirion (Sensirion, c). Fig 2.1 highlights the growing class of sensors that are being embedded into iPhones.

Thus, mobile devices are no longer only a means of communication. They provide an attractive platform for developing new and interactive applications to leverage the increasing sensing capabilities. By using an application programming interface, the sensors in the device can be exposed and manipulated to enhance various domains with this disruptive technology. The design of such sensing applications follows a common design pattern. First raw data is collected using the embedded sensors. Next, analysis is done on this data to infer

Fig. 2.1 Representation of the rise in number of embedded sensor in iPhones.

activities of interest or context of the data. Finally, high-level results are distributed to concerned communities and used to adapt future applications.
By engaging the human in the sensing and communication process, people-centric models have been presented for mobile sensing. In accordance to the awareness and involvement of the mobile device user, two contrasting sensing paradigms, namely participatory and opportunistic sensing, have been categorized (Lane et al., 2008). Participatory sensing (Dua et al., 2009; Restuccia et al., 2016) is an approach in which individuals and communities use these evermore capable mobile phones and cloud services to collect, publish and analyse sensed data. This involves active participation from the user-end involving decisions regarding the shared data. Involving the human in such significant decision stages, implies that the people carrying the devices become the sensor nodes, without the need of any other pre-installed infrastructure. In contrast, passive involvement or almost no human intervention gives rise to opportunistic sensing (Campbell et al., 2006, 2008). This paradigm allows automatic collection of sensed information from a mobile device in a required state by the application. This shifts the burden to the sensing system which is made more intelligent to automatically determine when the device meets the application request criterion and is in a position to capture interesting data. Both the paradigms empower people to collect and share sensor data across many applications. They help in providing a micro and macroscopic view of countries and individuals (Khan et al., 2013) by operating at three distinct scales, defined in the research community. These include:

1. Personal or Individual Sensing: This type of sensing is typically designed for a single individual. By collecting and analysing sensitive data regarding the mobile device user, these personal systems are used mainly for monitoring and sharing health and fitness related parameters, studying daily life patterns and enhancing personal social growth. Even behaviour intervention applications have been designed based on analysis of the data.

2. Community or Social Sensing: This type of sensing is based on individuals participating for a common concern or interest. Sensing information is collected within the social groups and may be related to achieving a group goal such as tracking neighbourhood safety, collective recycling efforts etc. This is also shared with other related communities and includes an inherent sense of trust.

3. Public Sensing: This type of sensing is focused at general public good and uses aggregation to minimize identification of individual patterns. Large number of people

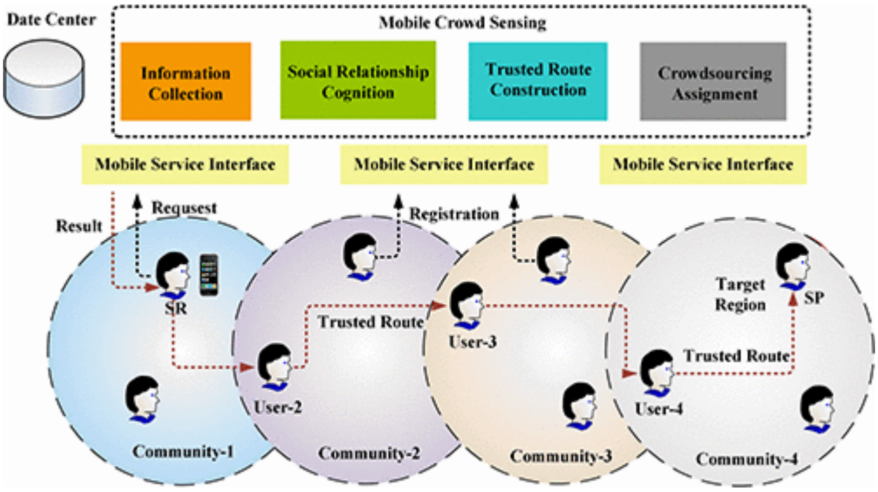


Fig. 2.2 A typical mobile crowd-sensing model as presented by Jian et al. (2015).

are encouraged to install such applications which range from tracking the spread of a disease, congestion or pollution across a city to fine-grained traffic information.

From embedded infra-red sensors to the touch-screens of mobile devices, these sensing systems play an instrumental role in the creation of various applications that harness this diverse available sensor data to influence daily life events. This includes a wide variety of domains, such as healthcare, environmental monitoring, homecare, social networks, safety, e-commerce and transportation (Khan et al., 2013). Multiple crowd-sensing platforms have also been presented, considering the mobility, sociality and complexity of mobile users (Guo et al., 2015; Jian et al., 2015) and incentives for engaging mobile users have also been presented (Zhang et al., 2016b). Fig. 2.2 shows a typical mobile crowd-sensing model as presented by Jian et al. (2015).

There are already plans to modify several exciting applications such as Google's StreetView, Senseable City Laboratory, MIT, and Intel's Urban Atmospheres project to work with mobile sensing, wherein opportunistically sensed data is made available via mobile devices. Lane et al. (2008) explore the case for the Google StreetView and argue that a variant using mobile sensing would allow a more frequent update of valid images and remove the constraints of travel to risky areas. Amongst many others, some existing sensing applications include StressSense (Lu et al., 2012), CarSafe (You et al., 2012), CenceMe (Miluzzo et al., 2007), NoiseTube and MobiShop (Khan et al., 2013). Several new research challenges regarding personal data security, analysis and classification of sensed data have also been introduced.

The main technical challenges in this field are presented below:

- **Energy Efficiency**: Due to applications increasingly using continuous sensing, the management of energy consumption is limited. The real-time processing of incoming sensed data jeopardizes the battery backup in mobile devices. Miluzzo et al. (2008), show that device standby time can reduce from 20 hours to 6 hours upon deployment of applications using continuous sensing strategies. Also the redundancy in incoming data is not checked which leads to unnecessary processing of similar data several times. Managing accuracy with energy efficiency becomes an important challenge. Ranging from system-level designs during continuous sensing to include low-power processors (Priyantha et al., 2011), bidirectional feedback pipelines (Kang et al., 2008), context correlation with association rules (Nath, 2013) and sensing pipelines (Lu et al., 2010), multiple techniques have been presented by researchers for this purpose. Furthermore, energy-accuracy trade-offs using adaptive sampling intervals and dynamic sampling frequencies (Ben Abdesslem et al., 2009; Lin et al., 2010; Rachuri et al., 2011; Sarker et al., 2016) have also been investigated.

- **Privacy**: Respecting the privacy and security of the personal data of the mobile user is a big technical challenge. How much information should be visible, what data reveals personal sensitive information etc. are still questions that need to be tackled. Researchers have introduced multiple techniques in answer to these questions, including weighted aggregation of encrypted data (Miao et al., 2015), token-based distributed systems (Krontiris and Dimitriou, 2015), and credit-based privacy-aware incentives (Li and Cao, 2016).

- **Phone Context**: The environment of the mobile device can be classified to discern various contexts. One important challenge here, is to understand the movement of mobile devices, as the devices might have limited access to the event and statistical methods might not be sufficient to generalize the context for unexpected environments. Additionally, it is important to reliably infer the wide-spectrum of human-activities using multiple reading from various sensors, under real-world conditions (Lane and Georgiev, 2015). Thus rich analytics must be performed on extracted features of the sensed data and methods ranging from distributed machine learning techniques (Miluzzo et al., 2010) and deep learning methodologies (Lane et al., 2015) to latent context identification (Unger, 2015) and hierarchical activity representations (Liu et al., 2016) have been proposed.

- **Dissemination**: Assuming that a particular effective mechanism is available that identifies context, the collected data must be made available to concerned communities and provide means of localizing the changed context. Thus dissemination of the sensed data at the right time to the right community is needed. Researchers have presented social distributed data dissemination algorithms (Xie et al., 2015a), data-fusion inspired cooperative forwarding schemes (Zhao et al., 2015) and public information tagging techniques for this purpose (Guo et al., 2015).

- **Programmability and Heterogeneity**: Although the number of sensors embedded in mobile devices is increasing, different vendors offer different interfaces to access the sensors on the mobile device and as such, their intrinsic performance differs across mobile devices (Cardone et al., 2016). Most of these interfaces just serve as black-boxes and there is no standardization in interfaces used across platforms. This makes it increasingly challenging to program applications. It is important to understand this heterogeneity as this can help in proper calibration of sensor data from multiple-resources (Li et al., 2015b).

2.1.2 Mobile Cloud Computing

Advances in current technology have already facilitated the migration of traditionally desktop-bound applications, such as word processors, and most other commonly used applications to cloud hosting in public or private cloud environments. In parallel, proliferation of smartphones and tablets means that these applications are being increasingly accessed from mobile devices. This presents a particular challenge for real-time interactive mobile applications requiring low latency.

This explosion of mobile applications, together with their rapid provisioning due to cloud computing, is motivating growing interest in Mobile Cloud Computing (MCC). This term was introduced not long after cloud computing came into existence and in its broadest sense refers to the application of cloud computing technology to the hosting and execution of mobile applications. From a resource management perspective, MCC can be viewed as an extension of the traditional data centre centric scope of cloud computing to encompass the provision of computational and storage resources by devices at the network edge. Fig. 2.3 shows the mobile cloud computing architecture as presented by Dinh et al. (2011).

These proposals range from the provision of computation and storage resources as an autonomous ‘data centre in a box’ at a wireless access router (Satyanarayanan et al., 2009) to creation of a ‘fog’ of virtualized computation, storage and networking resources provided by large numbers of network edge devices (Shi et al., 2012a). The former defines a cloudlet which acts like a one-hop, low-latency access to the cloud. This enables a rapid instantiation

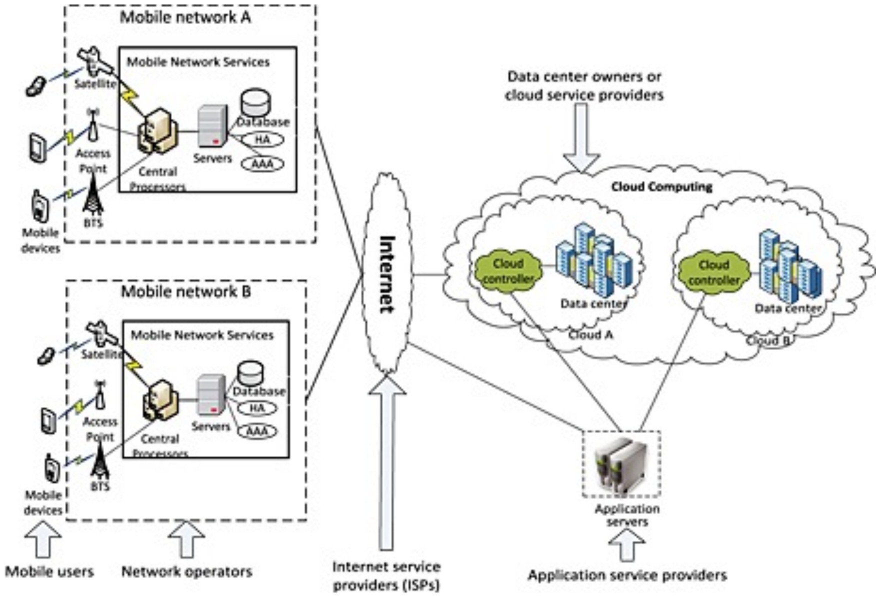


Fig. 2.3 Mobile Cloud Computing architecture as presented by Dinh et al. (2011).

of a service software that can then be easily utilized. In a world where cloudlets are installed like WiFi spots, these resource-rich attributes will help in personalized access along with meeting the demands for limited bandwidth. A mobile device can offload its data onto the cloudlet and make use of its computing power with better efficiency. It acts as a middle-ground between the cloud and the mobile device, enabling distribution of service requests and computation load onto the server. Fog computing envisions application logic and content placed in close proximity to the device users. Here, mobile devices themselves control the offloading of computation to other devices in order to minimize battery usage (Bonomi et al., 2012). This approach is seen as particularly useful for sensing applications, in which localized, short-term distributed computation can take place at the network edge and be supplemented by long-term, computationally intensive global analysis in a traditional data centre context.

MCC attempts to create a new paradigm for combining mobile web and cloud computing (Christensen, 2009). The powerful cloud servers are utilized for offloading storage and data processing operations enabling better performance (Abolfazli et al., 2015; Rahimi et al., 2014). As mobile applications continue to become more computation intensive, such offloading seems to present a viable solution and this technique saves energy significantly (Rudenko et al., 1998). However, total reliance on the cloud for computation can be more costly due to data transmission and limited bandwidth availability. Experiments have also shown that if the code is small, then offloading might consume more energy than local processing on the mobile device. Thus, an optimum balance needs to be achieved for efficient offloading and decisions regarding the amount of data that should be offloaded are to be made. The amount of energy saved in this procedure can be determined using the wireless bandwidth, the amount of computation and the data transmitted (Kumar and Lu, 2010).

As this information changes dynamically, these decisions need to be taken at run-time which has been critically analysed by Shiraz et al. (2015) and increases the complexity of the system. From cost-graphs, call graph-based models (Kaya et al., 2016), comparison of local and remote execution time to finding optimal partitions, lots of ideas have been proposed and studied for creating optimum task migration or offloading schedules (Liu et al., 2015b; Zhang et al., 2016a). Two basic partitioning methods are used, namely static partitioning and dynamic partitioning (Jagtap et al., 2014), but most frameworks rely on a combination of both methods. From self-cloning agent models (Angin et al., 2015) and machine-learning based offloading techniques (Eom et al., 2015) to the use of adhoc clouds using IoT devices (Deshmukh and Shah, 2016), several methods have been presented by researchers. The inability to accurately judge execution time correctly makes way for another algorithm (Xian et al., 2007), where a two-competitive method is introduced, optimized using online statistics to compute optimal time-out after which the computation is sent to the server. Researching talk about re-offloading the failed subtasks, periodic checking of connection as a prevention technique, task control flow graphs, markov decision processes for offloading (Terefe et al., 2016) and even game-theoretic approaches (Cardellini et al., 2016). Adaptive offloading considering context-based offloading (Zhou et al., 2015) and feedback loops (Amoretti et al., 2016) have also been presented.

Notable techniques for optimized offloading, making a trade-off between the local processing costs and the remote transmission of data between the cloud servers and mobile devices, are MAUI (Cuervo et al., 2010), CloneCloud (Chun et al., 2011) and ThinkAir (Kosta et al., 2012). MAUI, is a system that uses the benefits of a managed code environment to maximize energy savings with minimal burden on the programmer (Cuervo et al., 2010). It decides at run-time which methods should be remotely executed, driven by an optimization engine that achieves the best energy savings possible under the mobile device's current connectivity constrains. Two versions of the code run, one locally and the other remotely and serialization is used to determine network costs. CloneCloud (Chun et al., 2011) is useful for cloning the entire set of data/applications from the smart-phone onto the cloud. Only a 'right' portion of the application gets executed in the cloud in this scenario where it is significantly faster and the cost of sending/receiving the data proves to be worth it. However, it is limited in the migration of the native state and resources which remain unavailable for access. The system is a flexible application partitioner allowing calculated portions of the execution of services to offload into device-clones. Threads of the execution continue to migrate between the mobile device and the clone to maintain concurrency until it merges into the complete process. On the other hand, ThinkAir provides online method-level offloading in comparison to CloneCloud, thereby improving the restrictions that are placed in this scenario. It further improves scalability by exploiting smart phone virtualization. Fig 2.4 makes a comparison between these three offloading methods.

Applications supported by MCC such as M-Commerce, M-Learning, Mobile-Healthcare (Hoang and Chen, 2010) and Mobile Gaming (Dinh et al., 2011) are promising solutions for a better and more effective lifestyle (Wang et al., 2015). The user-centric security and privacy protection along with individual and collective sensing capability makes this field of interest to several researchers. Applications have also been centred on service searching including keyword-based, voice-based and tag-based searching. In addition there is a mobile-cloud collaborative application to detect traffic lights for the blind (Angin et al., 2010).

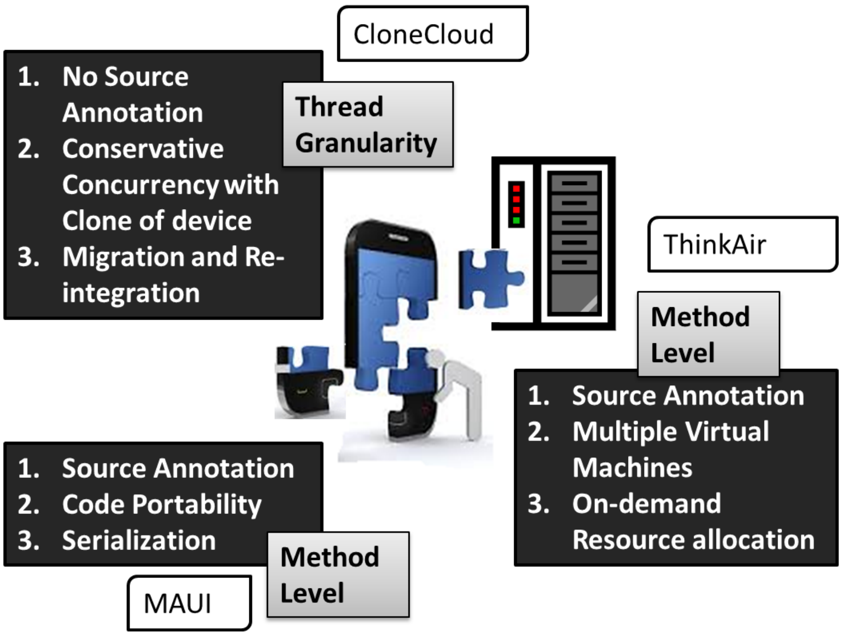


Fig. 2.4 Comparison of the methodology adopted by three notable dynamic offloading techniques, namely MAUI, CloneCloud and ThinkAir.

Mobile cloud computing, thus introduces a greater degree of resource heterogeneity, intermittent and highly variable resource connectivity and availability, as well as the competing management objectives. This clearly increases the number of challenges in this research field and the main challenges are presented below.

- **Network Latency**: Challenges need to be faced due to the intrinsic nature of the mobile networks (Dinh et al., 2011) as mobile device connections to the cloud suffer from high network latency and huge transmission power consumption, especially when using cellular technology. Additionally, mobile applications are becoming more latency-sensitive, but these networks require longer execution time for applications (Guan et al., 2011). Thus, latency needs to be effectively managed for better interactive experiences and researchers have presented cloudlets (Gai et al., 2016; Jararweh et al., 2014; Satyanarayanan et al., 2009), state approximations, event time-shifting (Lee et al., 2015), and pre-fetching techniques (Ko et al., 2016). Researchers have also studied the effect of latency on job completion rate while changing the scale of cloud network (Amoretti et al., 2016).

- **Limited Bandwidth**: This is one of the biggest issues in MCC. Researchers have discussed sharing the limited bandwidth among people to adequately utilize the resource. A coalition game is used to model users from the same area and requiring same content. Jin and Kwok (2010) derive that energy supply and system information about peer mobility are the foremost factors for coalition formulation. Similar work has been done by Guan et al. (2014) who focus on sharing content-streams in a peer-to-peer fashion. Jung et al. (2010) have discussed a data distribution policy to determine how much of the available bandwidth is allocated to a particular user. They propose a decision framework called RACE (Resource Aware Collaborative Execution) as a Markow Decision problem to protect the network capacity while allowing offloading of load into other networks. By studying bandwidth shifting and redistribution, Misra et al. (2014) present a utility maximization algorithm that guarantees quality of service using auction theory. Furthermore, Sun and Ansari (2015) present a bandwidth-allocation strategy for LTE networks which introduces a channel condition penalty function to improve energy-efficiency.

- **Wireless Connectivity**: The mobile network may be lost due to traffic congestion, network failure or out-of signal services (Dinh et al., 2011). However, it is imperative for MCC to have an available wireless link with scalability features. Also, the network should be energy and cost effective (Chetan et al., 2010). Researchers have talked about allowing a user to connect to the cloud through neighbouring stable or moving nodes in the vicinity. Researchers have studied improving energy-efficiency by adaptive network transmissions (Liu et al., 2015a; Mtibaa et al., 2013; Shi et al., 2012b), mobility-driven service provisioning (Li et al., 2015a) and opportunistic mobile computing methodologies (Chatzopoulos et al., 2016; Mtibaa et al., 2015) to exploit nearby computational resources.

- **Security and Privacy**: It is commonly known that users lose their mobile phones. A security breach might occur if such a device contains valuable information downloaded minutes ago (Cox, 2011). Also, it is difficult to enforce protection mechanisms in mobile devices due to the heterogeneity (Guan et al., 2011). Thus, handling security and privacy concerns is a major challenge (Naik and Jenkins, 2016). Researchers have presented various authentication mechanisms for mobile cloud computing ranging from fuzzy vaults and digital signatures to context-based credentials (Ali et al., 2015; Alizadeh et al., 2016).

- **Heterogeneity**: The wide range of heterogeneous operating systems and devices increases the complexity of MCC (Dinh et al., 2011). The compatibility of the different mobile network interfaces and standards further increases these challenges (Lei et al., 2013). A comprehensive survey on this has been presented by Sanaei et al. (2014), who analyse and classify the heterogeneity present in mobile cloud computing. Recently, Han et al. (2015) have discussed a 5G network infrastructure with autonomous radio access technologies to enable management of multiple network interfaces.

2.1.3 Mobility Models

The collaborative sensing framework uses sensed data from mobile devices that are carried by humans and as such, it is imperative to accurately model movement of the devices. For this purpose, this subsection presents the mobility models defined in current literature. Most early studies regarding mobility models were done for Mobile Ad-Hoc Networks or MANETs to see the performance of MANET routing protocols. It was described as the pattern that mobile users follow with changes in location, velocity and acceleration. The idea was to create a realistic model to infer accurate information. Various researchers have proposed mobility models with underlying unique features to emulate a real-time movement and categorized them accordingly. One classification distinguishes between mobility models

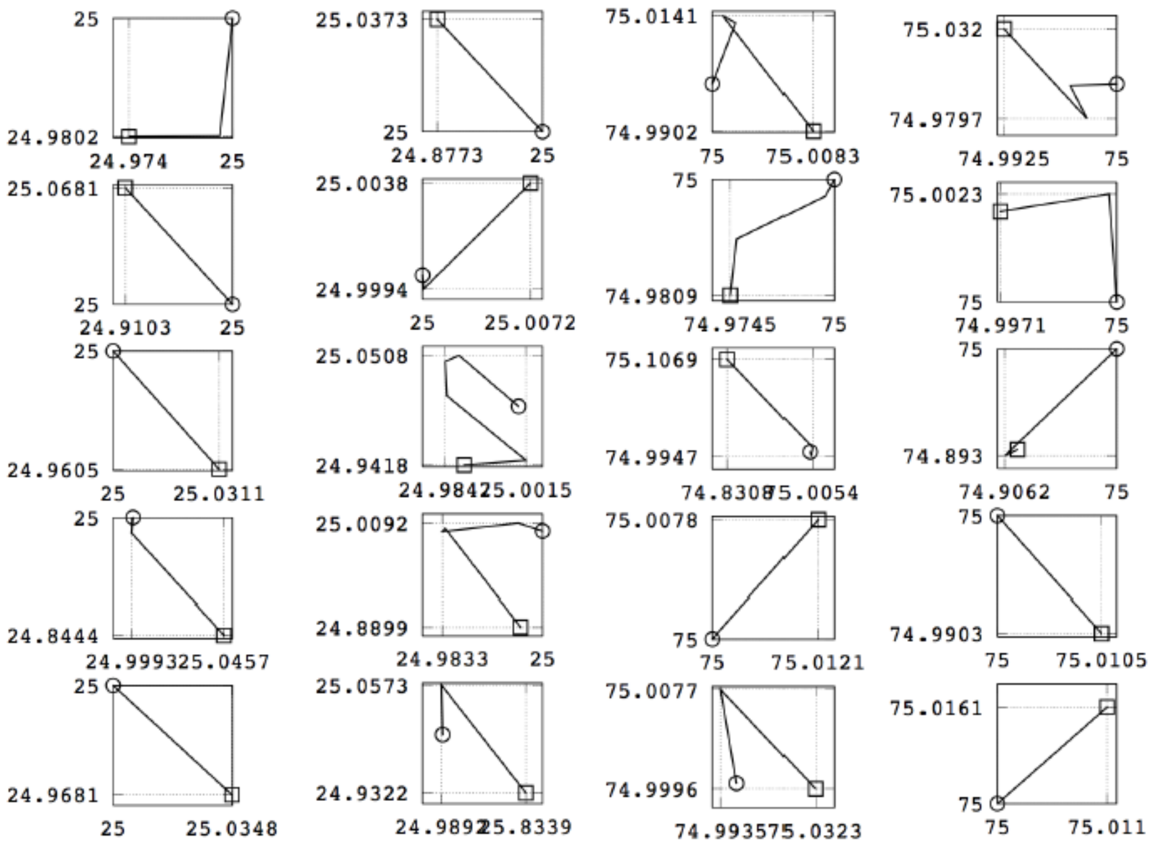


Fig. 2.5 Simulated Truncated Levy Walk mobility model for ten instances of mobile devices, starting at $[25,25]^T$ (columns 1,2) and $[75,75]^T$ (columns 3,4) for two hours. The start locations are denoted by circles and the end locations are given by squares.

on the basis of dependence on other mobile users (Camp et al., 2002) and broadly discerns two groups. These are the entity mobility models (mobile devices move independent from each other) and the group mobility models (mobile devices move in groups or are dependent on a single leader).

Another classification is based on mobility characteristics (Bai and Helmy, 2004) which has updated and extended into five subcategories by Aschenbruck et al. (2008). The first category contains random mobility models which have no restrictions in terms of memory or movement and are most commonly used in simulations. These include the Random Waypoint model and its variations such as Random Walk and Random Direction. Models incorporating movement history (Gauss-Markov and Smooth Random Model) and those with correlated group movement (Reference Point Group Model and Structured Group Mobility Model) fall under the next categories of models with temporal dependencies and

spatial dependencies respectively. If a model is restrictive in terms of the area (Graph Based Mobility Model and Manhattan Grid), it is classified under models with geographical restrictions. The last category includes all mobility models which have hybrid characteristics (Freeway Mobility Model and User-oriented Mobility Meta-Model).
However, these models are unable to reliably capture the movement of humans. Although probabilistic models for user movements have been adopted by researchers (Lin et al., 2010), in recent times, human mobility is more widely studied as a Levy walk model. This is generalized from the early studies regarding animals like the albatross or monkeys which followed a power-law distribution. González et al. (2008) have shown that human mobility can be compared to a Levy walk with heavy-tail flight distribution by using anonymous mobile device traces as an extension to the work of Brockmann et al. (2006) who used data from bank notes in circulation. A Levy walk intuitively reflects a human mobility pattern as its consists of more short displacements than long displacements. Their results show how a long tailed distribution captures this aspect along with highlighting the probability of humans to frequent locations. The extension to include visiting times and frequency by Song et al. (2010) presented a 93% predictability factor regarding human mobility with data history. Rhee et al. (2011) have presented a truncated Levy walk model which provides a simple and realistic model for human mobility. Fig. 2.5 depicts the simulation of ten instances of mobile devices, starting at two sets of coordinates $\mathbf{x}_i(0) = [25,25]^T$ and $\mathbf{x}_j(0) = [75,75]^T$, and running for a period of two hours following the model. This model has been incorporated into the collaborative sensing framework and more details are present in the following chapters.

2.2 Literature Review

This section presents a literature review of collaboration models, aggregation techniques, context-awareness with energy-efficiency and efficient cluster-head selection. These form important aspects for the collaborative sensing framework and help lay a foundation for the current and ongoing research.

2.2.1 Collaboration Models

A focal point of the collaborative sensing framework described in this dissertation, is the proposal of cooperating mobile devices that sense data from their surrounding and exploit the scalability and processing capability of the mobile cloud. This becomes specially challenging due to the heterogeneity of application requirements, mobile operating systems, network interfaces and other mobile device capabilities. Several collaborative methodologies and algorithms have been presented by researchers, from studying profit margins for service providers to enforcing privacy and data integrity, which are presented below.

Harnessing the portability of mobile devices and wide spread of 3G/4G networks and WiFi accesses, iCoMe (Shah-Mansouri and Wong, 2014) is one such incentive-based cooperative resource management technique that focuses on increasing the revenue of the service provider. Furthermore, Tang et al. (2016) consider a broker-based mobile cloud to present a double-sided bidding mechanism for resource sharing whereas other researchers focus on participation incentives for mobile users to collaborate (Feng et al., 2014; Gao et al., 2015). Several other resource managements systems have also been presented that allow mobile devices to actively download data from available cellular and WiFi connections (Kaewpuang et al., 2013; Shu et al., 2013). A cloud-on-the-fly approach for collaboration between devices also proves to show promising results and is termed as Transient Cloud (Penner et al., 2014).

The concept of cyber-foraging has also encouraged collaboration to exploit nearby mobile resources (Liu et al., 2015a; Shi et al., 2012b) and opportunistic mobile computing methodologies (Chatzopoulos et al., 2016; Mtibaa et al., 2015) have been presented. Xiao et al. (2016) present a delay-sensitive, sensing-duration-aware algorithm in which mobile users cooperate to ensure that the deadline is met. Other parallel offloading techniques to speed computation and increase device lifetimes have also been proposed. These further examine how the tasks can be scheduled and allocated to the surrounding resources, since the network connection might be intermittent (Mtibaa et al., 2013). Use of a non-transferable coalition formation game setup, taking into account quality-of-service (QoS) requirements and spectrum utilization, has also been presented (Wu et al., 2014). The concept of using multiple access links for collaborative downloading and gateways has also been proposed with promising results (Ananthanarayanan et al., 2007).

Privacy and threat issues present in such collaborative systems for large scale mobile cloud have also been studied. In AnonySense (Cornelius et al., 2008; Shin et al., 2011), the principle of anonymized data source has been presented whereas the use of personal data vaults has also been proposed (Mun et al., 2010). Furthermore, Ravichandran et al. (2015) leverage privacy annotations and restrict access to enable a transparent execution migration framework. Privacy policy enforcement using programming model for adherence have also been explored by Yang et al. (2012).

2.2.2 Aggregation Techniques

In the research presented in this dissertation, it is proposed that the capabilities of the embedded sensors and mobile devices are aggregated and multi-tasked by the collaborative sensing framework, for improved performance and energy-efficiency, when sensed data is transmitted to multiple applications. Several aggregating methods are available for this purpose, and it is challenging to determine the best technique given the temporal and spatial complexities and constraints of the problem.

In this context, several data aggregation techniques have been extensively studied for both energy efficiency and congestion reduction. Examples include cluster-based heuristics (Dasgupta et al., 2003; Ranjani et al., 2012; Velmani and Kaarthick, 2015), data compression (Baek et al., 2004; Xiang et al., 2011), hierarchical aggregation (Chen et al., 2006; Xu et al., 2015), entropy-analysis (Galluccio et al., 2008) as well as other distributed data aggregation techniques (Jesus et al., 2015). Different alternative models are also available that use tree selection methods including Breadth-First Search (BFS), Depth-First Search (DFS), Flooding, Well-Connected Dominating Set (WGDS) and their modified forms (Fotue et al., 2010).

For mobile sensing, several privacy-preserving methods of aggregation are also available which include homomorphic encryption (Li and Cao, 2013; Li et al., 2014) with blind signatures (Fan et al., 2015) and erasure coding technology for slicing data to maintain anonymity (Xie et al., 2015b). Zhuo et al. (2016) have also presented a data aggregation technique where these tasks are allocated to the cloud for a mobile crowd-sensing scenario, in contrast to Zhang et al. (2013) that present a peer-to-peer based scheme for people-centric urban sensing. An incentive, data aggregation, and data perturbation mechanism, called 'Inception' has also been introduced to support data aggregation in mobile crowd-sensing scenarios that also ensures data privacy (Jin et al., 2016).

Frequent Patterns have also played an essential role in data mining tasks that aim at aggregation by extracting interesting patterns from databases, such as association rules, clusters, classifications, correlations and sequences (Han et al., 2006, 2007a). They have been also widely employed in recommendation systems, web mining and software bug mining (Han et al., 2007b). Web searches use similarity space metrics to represent similar pages to the user (Zobel and Moffat, 1998) such as vector space models, Pearson-correlation model, as well as Bayesian classifiers (Su and Khoshgoftaar, 2009). It has also been extended to sequential pattern mining and found applications from extracting patient paths, dyspepsia symptoms to patterns in group activity, task and resource sequences (Kumar et al., 2011). In healthcare and medicine, problems such as genome analysis, drug design and risk patterns mining (Jung et al., 2015; Li et al., 2005) have been addressed with frequent patterns. Frequent pattern mining has recently also been used for activity recognition (Wen et al., 2015), predictive caching technologies (Dutta et al., 2015) and malware detection (Fan et al., 2016).

2.2.3 Context-Awareness with Energy-Efficiency

Most mobile-sensing applications contain a degree of context-awareness to improve personalization and provide their users with an enhanced experience. Thus, the collaborative sensing framework focuses on incorporating algorithms for detecting, localizing and exploiting this context-awareness whilst maintaining energy-efficiency. This becomes challenging under real-world conditions, as mobile devices might have limited access to the context changing event which makes it difficult to reliably infer from the wide-spectrum of human activities. Additionally, battery backup of the device is jeopardized when continuous sensing strategies are used to update the context surrounding the device user.

Other researchers have also shown interest and presented methodologies to benefit from this context information for mobile sensing applications. Comprehensive survey work has been provided by Lane et al. (2010), Khan et al. (2013) and Campbell et al. (2006, 2008). Furthermore, context-awareness is also used for various other purposes, including QoE prediction (Mitra et al., 2015), adaptive privacy (Schaub et al., 2015) and energy-aware selection criterion for a predefined sensing tasks (Marjanović et al., 2016; Yürür et al., 2015). A widely applied example is the use of GPS sensors, often in combination with WiFi positioning, to locate devices (Ye et al., 2012). Other localization algorithms have been specified by researchers for retrieving sensor locations in wireless sensor networks (Mao et al., 2007) and in odour detection (Ishida et al., 1996)—in the latter case, mostly via probes or robots. In wireless sensing, angle-of-arrival measurement, distance related measurements and RSS profiling techniques are used.

Multiple sensors have also been employed to create activity recognition applications (Anjum and Ilyas, 2013), detect fatigue in drivers (Tu et al., 2016) and motivate physical activity (Gupta et al., 2016). Projects like BikeNet (Eisenman et al., 2010), Ear-Phone (Rana et al., 2015) and EMC (Chen et al., 2015) use the capability of the sensors surrounding the human to send context feedback for assistance while performing tasks. The integration of sensors surrounding the user other than mobile sensors has been also presented in the OPPORTUNITY framework (Kurz et al., 2011). A game theory model called

GlobalSite (Rahmes et al., 2013) has also been suggested with multiple contextual information for threat analysis.

With regards to environmental monitoring, specifically fire detection (Giglio et al., 2003) and odour detection (Ishida et al., 1998), multiple techniques are investigated in literature. Wireless sensing platforms are developed for indoor detections (Purohit et al., 2011) and weather predictions (Phillips and Sankar, 2013) have been improved with opportunistic sensing by using social media platforms, but mobile devices have not been used. However, multiple mobile crowd-sensing applications have been studied (Ganti et al., 2011; Jian et al., 2015).

As context-aware applications become more computation-intensive, it becomes important to reduce the energy consumption during this process. Efforts to make mobile sensing systems more energy-efficient range from low-power processors (Liaqat et al., 2016; Priyantha et al., 2011) and sensing pipelines (Lu et al., 2010) to bidirectional feedback (Kang et al., 2008) and context correlation with association rules (Nath, 2013). Energy-accuracy trade-offs using adaptive sampling intervals and dynamic sampling frequencies (Ben Abdesslem et al., 2009; Lin et al., 2010; Rachuri et al., 2011; Sarker et al., 2016) have also been considered, with additional techniques such as mobile tethering with cloud gatherers and an energy-aware striper (Sharma et al., 2009), being incorporated. Furthermore, multiple access links for collaborative downloading (Ananthanarayanan et al., 2007) and data pre-fetching methods (Bharath, 2014; Wang and Chen, 2014) have also been investigated.

2.2.4 Cluster-Head Selection

One method of further improving the efficiency of the collaborative sensing framework is by optimizing the selection of mobile devices to offload sensed data. Clustering algorithms has been proposed in this dissertation to decide which mobile devices must be given the responsibility of becoming a cluster-head to collect and transmit the sensed data. This can be challenging since the algorithms need to fairly share transmission overheads between the mobile devices in an ad-hoc, low latency, bandwidth and energy efficient manner.

Such selection of one or more devices from the set of collaborating devices has been widely studied in wireless sensor networks. The most widely known routing protocol that stochastically selects cluster-heads is LEACH (Heinzelman et al., 2000). In recent years, many clustering protocols have been adapted from the underlying threshold framework of LEACH to improve network lifetime and energy-efficiency (Ramesh and Somasundaram, 2011) with mixed results. Factors like residual energy (Handy et al., 2002; Razaque et al., 2016; Thein and Thein, 2010), distance to the base station (Kang and Nguyen, 2012; Sharma et al., 2015), centralized algorithms with location information with the base station (Heinzelman et al., 2002) and optimal stabilization of number of cluster-heads (Batra and Kant, 2016) have been used to study their effect on cluster-head selection. Aldeer et al. (2016) also study optimal placement to minimize maintenace cost of clusters. Semantic Clustering Models with fuzzy inference systems have also been introduced (El Alami and Najid, 2016; Rocha et al., 2012).

Furthermore, bridging the gap between communication and biological systems, bio-inspired solutions have also been presented by researchers (Dressler and Akan, 2010) for selection of one device to send sensor data. These include centralized approaches in BEE-C (da Silva Rego et al., 2012) based on the honey bee's mating behaviour, Particle Swarm Optimization (Latiff et al., 2007) and distributed T-ant (Selvakennedy et al., 2006) based on ant colonies. The Artificial Immune System (AIS) (Hofmeyr and Forrest, 2000) has also been inspired from the mammalian immune system to use its self-learning and memory characteristics for better selection decisions. These AIS algorithms have been implemented for data mining, data analysis (Timmis et al., 2000), pattern recognition (Wang et al., 2008) and anomaly detection. Atakan and Akan (2006) have presented an immune system based node selection technique for wireless sensor networks. Here cell simulation due to a pathogen in the immune system is modelled as designated node selection (similar to the cluster-head) from the sink. Nodes are selected based on correlation coefficients depending on their distance from the sink and surrounding neighbours. This is analogous to the wireless sensor network and mobile sensing scenario.

2.3 Summary

This chapter presented a review of the technologies, trends and current literature that are supplemental to the research proposed in this dissertation and laid a foundational base for effectively building the collaborative sensing framework. It commenced with a background study on mobile sensing and mobile cloud computing, including definitions, algorithms and challenges faced within these fields. It also described mobility models that are available to simulate human movements. Although some works have already advanced development, research issues are still left to be addressed. These help in streamlining the research that has been presented in the latter chapters and have been incorporated in the research questions presented in Chapter 1. Next, a literature review on collaboration models, aggregation techniques, context awareness and cluster-head selection was presented. These are all related works for the collaborative sensing framework and help in identifying the areas of contribution of this dissertation.

As presented above, most researchers have focused on selecting parts of the applications to be offloaded into the cloud server in mobile cloud computing or exploit nearby resources for improved efficiency. However, limited researchers attempt to create an abstract layer interposed between the various sensors, mobile devices and multiple applications to organize **where** the sensing must occur and construct a sensing schedule. Furthermore optimizing decisions regarding **how many** mobile devices are sufficient to offload the data with pre-requisite accuracy has not been attempted, specially for multiple applications. These challenges have been formulated as part of the first (RQ1) and second (RQ2) research questions. In contrast to the collaborative models and data offloading platforms, this dissertation considers multiple applications requesting sensed data and devices interacting with a centralized sensing framework for aggregating the sensed data streams and sending data into the cloud for processing. The capability of a single mobile device is multi-tasked to reduce redundancy in sensed data for more than one application. For this purpose, a revised version of the frequent pattern growth algorithm, an unsupervised learning technique is used which, to the best of knowledge, has not been used for aggregation in mobile sensing and mobile computing. Details of this work are presented in Chapter 3.

Another important challenge is deciding **when** a mobile device should sense and report the data, in an energy efficient manner. Considering the surrounding IoT devices, the framework must also be in a position to decide **what** device (an IoT device or mobile device) should be given this responsibility. These challenges have been formulated as part of the third (RQ3) and fourth (RQ5) research questions. Multiple context-aware algorithms have been presented in this respect that perform operations based on data inference, with limited recognition of the event-driven nature of the applications. In contrast, this dissertation presents the idea of adopting a state-machine driven approach to leverage application-specific knowledge for collaboration amongst devices when offloading sensor data. Energy-efficiency is further achieved by using multiple sensors from the same mobile device for multiple applications instead of the low-power, low-accuracy or sensing pipelines approach in literature. This further reduces the number of mobile devices that are actively losing energy in the environment. Furthermore, energy is also saved by reducing the offloaded volume of the sensed data by intelligent reporting supported by the state machines to understand when the context demands for data offload.

The last and fifth research question (RQ5) of this dissertation focuses on **who** is optimally placed to offload sensed data. This is an important challenge and relates to deciding the best mobile device which can be selected as the cluster-head to offload the sensed data. A similar approach to the stochastic cluster-head selection models, presented in this chapter is used, but the non-static nature of the mobiles introduces other challenges. In contrast to the other approaches, location information is used to optimize the trajectory of the cluster-heads. Neighbouring mobile devices bias their selection probability relative to a well-positioned cluster-head to ensure that the responsibility passes from a well placed position to a better one successively, and this bias is forgotten after a suitable amount of time has passed. Furthermore, another difference from the previous approaches is to adapt the number of clusters created by understanding the demands of the sensing scenario.

Thus, this chapter highlighted how current literature is unable to address the research questions defined for this dissertation. It further helped to compare and contrast between the proposed research and existing work.

Chapter 3

Sensing Framework with an Aggregation Model

3.1 Introduction

The proliferation of smart mobile devices, having multiple sensing capabilities and significant computing power, enables their inclusion into mobile sensing systems. However, major concerns for such systems are the resource and processing limitations of mobile devices. The issues of mobile communication combined with the highly heterogeneous landscape in terms of wireless network interfaces, further adds technical challenges. Although multiple data offloading techniques have been proposed in mobile cloud computing to reduce computation and improve energy-efficiency, it is imperative to design an abstract layer interposed between the sensors, mobile devices and applications that endorses coherent interaction, supports their collaboration and maintains content integrity. The first question (RQ1) presented in Chapter 1 of this dissertation collates all these issues.

Moreover, another key concern with such a system is the need to balance accuracy of the information received by the application and the volume of data offloaded by the energy-restricted mobile devices. Since multiple mobile devices are actively participating, it often amounts to overlapping streams of redundant sensor data being offloaded, leading to a wastage of resources. Also, increased network traffic due to the large numbers of mobile devices offloading to the cloud may lead to processing and latency issues. Thus, another important technical challenge is the optimal selection of sensors from mobile devices to accurately satisfy application constraints in an energy-efficient manner while reducing the number of active devices. This difficulty is exacerbated by the fact that handsets are typically mobile, so the sensing system needs to embody human mobility models. These challenges have been presented in the second research question (RQ2) in Chapter 1 of this dissertation.

This chapter details the first body of work completed as part of this research study and addresses the above stated questions by presenting a centralized sensing framework to support sharing of resourcing between co-located mobile devices. It embodies a middleware architecture that receives all the application requests and requirements, communicates with the devices and sensors present within a specified physical sensing area and provides an scheduling methodology to ensure that all requirements are met in the most energy-efficient and seamless manner. It also seeks to identify the best combination of sensors to activate on the available devices to create an aggregation model. Additionally, the chapter focuses on providing a methodology for aggregating sources of data so that battery life of mobile device is more efficiently used. This is achieved with Algorithm Info-Aggregation, running on the sensing framework that uses a novel frequent pattern mining approach to reduce the number of active mobile devices in a collaborative sensing system so that data served by a given device can be availed by multiple applications.

This chapter is organized as follows. First, the centralized framework is introduced along with a description of the scenario that has been considered. Next, a mathematical formulation of the trade-off between the volume of the sensed data received by applications and the energy required to transfer data offloaded by the active mobile devices is presented. This takes into account constraints related to device mobility, residual energy levels and multiple application requirements in terms of the physical locations to be covered and frequency of data reporting. The next subsection presents the collaborative sensing scheme that seeks to multitask the capabilities of a single device with Algorithm Info-Aggregation to maximise the degree to which sensed data transferred from a given mobile device can be served to more than one application. The performance of this algorithm is then compared in the next section with Algorithm No-Aggregation that does not recognize the potential to reduce redundancy in the offloaded data. This evaluation section presents results for the simplifying case that mobile devices are static and also incorporates a mobility model into the simulation study for a complete and more realistic analysis of the sensing framework. In either case, the algorithm shows improvement in terms of energy utilization of the mobile devices whilst reducing the number of active mobile devices and the volume of offloaded sensed data. This chapter concludes by presenting a summary of the research and its applicability with respect to the research questions. This work has been disseminated in Loomba et al. (2014b) and Loomba et al. (2015b).

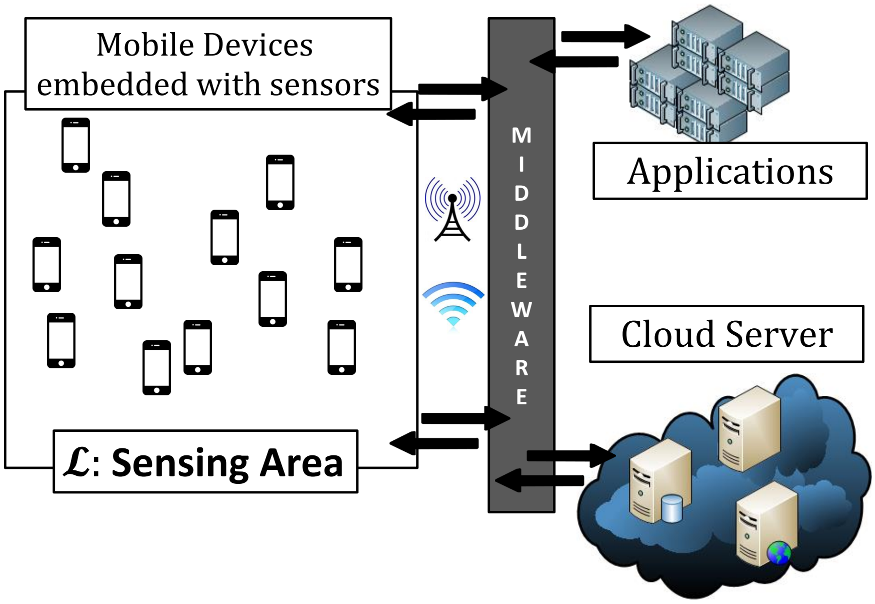


Fig. 3.1 Centralized Architecture Representation of the Collaborative Sensing Framework, facilitating seamless interaction between the mobile applications, mobile devices with embedded sensors and cloud infrastructure in a square sensing area.

3.2 Centralized Architecture for Collaborative Sensing Framework

An important feature of mobile sensing that distinguishes it from traditional sensor networks is the absence of single data ownership. This framework is tailored to suit both sensing paradigms, namely participatory and opportunistic sensing by acting as a collection point for sensed data. In participatory sensing, mobile users manually select the sensors that would be available for public access and provide paths to activate the embedded sensor-types. In opportunistic sensing, the control is with the framework to access any/all sensors embedded into the mobile device when such a state may arise.

Such a system must also be able to manage content integrity whilst providing a platform for seamless interaction between the mobile devices. However, privacy and other security issues are not within the scope of this work, which mainly focuses on creating an energy-efficient sensing framework. But the framework considers anonymized sensed data collection and transmission in a centralized manner which makes it amenable to preserve privacy and protect sensed data from other mobile devices participating in the network.

The framework is planned as the first point of contact for the plethora of mobile sensing applications, ranging from monitoring environmental factors like temperature, pressure, light, humidity, to health monitoring and promoting social interactions. Envisaging pico-cell deployment (with a range up to 200m) in urban cities(§3.3 and §4.3.1) or enterprise industrial deployment (§4.2) or even sub-urban deployments in open areas like parks and railway stations (§5.2), the potential of hundreds of mobile devices is exploited to collaborate and gather sensor data for the benefit of these applications. Depending on the requirements to mediate between the mobile devices and the applications, the framework becomes responsible for periodically determining which mobile device to activate for sensing, processing and transmitting data. Here, it is assumed that mobile device users do not have any malicious intent and provide accurate data. This centralized architecture also means a reduction in coordination messages among the users which is a desirable quality for such a system.
This setup is considered since it juxtaposes centralized coordination, to compute an aggregation and scheduling solution, with decentralized execution, when each mobile device senses and reports anonymized pre-processed data. For smaller deployments, this data is then sent to the cloud server for further processing and storage and made available to the application. For crowd sourcing applications and large-scale deployments in cities, these small deployments can be interpreted as local clusters of mobile devices that use the centralized framework as a gateway to communicate and aggregate data. This leads to a distributed scheme where sensor data from various sources is available to the applications depending on their requirements. As such, this can facilitate the creation of an aggregation hierarchy where different levels follow different aggregation mechanisms allowing low-level to highly processed sensed data being made available to the applications. This chapter presents a small deployment solution with hundreds of devices/applications with sensing data that does not change rapidly. This current scenario has been presented in Fig. 3.1. Large scale deployments with hierarchical aggregation structures will be considered as future work of this dissertation.

3.3 Problem Formulation

This section presents a mathematical formulation to optimally utilize knowledge of mobile resources and application requirements to select the minimal number of mobile devices that can be used to gather sensor data, subject to constraints relating to coverage area, sensor data sampling rates and residual battery. Fig. 3.2 presents the mobile sensing model that has been

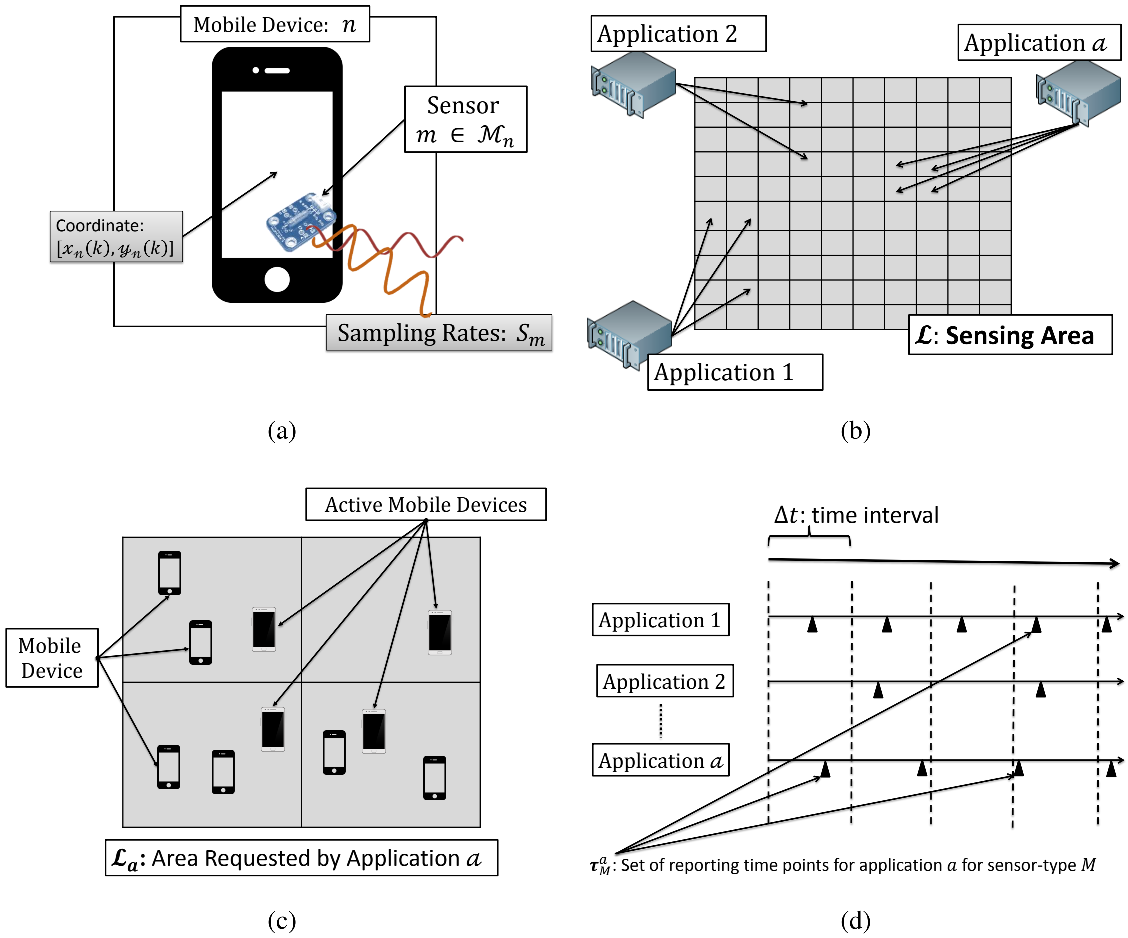


Fig. 3.2 Mobile Sensing Model for the collaborative sensing framework: (a) shows a mobile device, at coordinates $(x_n(k), y_n(k))$, with an embedded sensor having different sample rates for time interval k; (b) shows applications defining different request locations $\mathscr{L}_a$ to be covered within the sensing area; (c) shows different mobile devices covering the area requested by an application at one time instant; (d) shows the reporting time points of the various applications in the time intervals for the problem formulation.

considered for this formulation and all mathematical notations have been defined in Table 3.1.

3.3.1 Terminology

This problem is formulated over a total duration of $\mathscr{T}$ seconds, which is subdivided into smaller time intervals. The duration of these time intervals is determined by the smallest time period over which sensed data is required by the set of applications and is defined as Δt

seconds. For $t \in \mathscr{T}$ and $k \in [1, \mathscr{T}/\Delta t]$, time interval k is thus denoted by $(k-1)\Delta t < t < k\Delta t$.

The physical area of interest to the applications is represented by $\mathscr{L}$, and specified as a set of square request locations that are covered by mobile devices and can be requested by individual applications. The request location i is defined as $\mathscr{L}_i \in \mathscr{L}$. $\mathscr{N}$ denotes the set of mobile devices located within the area of interest, and each device $n \in \mathscr{N}$ follows a mobility pattern with its trajectory limited within its boundaries. The 2D coordinates of the device are used to identify its position within these request locations and defined as a function of the time interval k to be $(x_n(k), y_n(k))$. It is assumed that accurate sensor information can be sent by the mobile device n when active for the square request location in which it is present for time interval k. Additionally, $\mathscr{N}_i^k$ represents the set of all devices present in a request location, mathematically represented by $\forall n \in \mathscr{N}_i^k, s.t. (x_n(k), y_n(k)) \in \mathscr{L}_i \in \mathscr{L}$ for time interval k.

Each mobile device has embedded within it different sensor-types and $\mathscr{H}$ is the set of all sensor-types for available mobile devices. Each mobile device n is defined to contain $\mathscr{M}_n \subseteq \mathscr{H}$ sensor-types with an individual sensor-type being represented as $m \in \mathscr{M}_n$. Each device is also capable of providing multiple sampling rates in the set $\mathscr{S}_m$ for each of its sensor-types. Additionally, the volume of sensed data accumulated by a device is based on the sampling rate and sensor-type. The variable v_m^n is defined as the volume of sensed data collected by sensing activities of sensor-type $m \in \mathscr{M}_n$ on device n whenever it is activated. Fig. 3.2a depicts the properties of an embedded sensor in a mobile device.

An application $a \in \mathscr{A}$, where $\mathscr{A}$ denotes the set of applications, is defined with potential interest in data from sensor-types $\mathscr{M}_a$ during $\mathscr{T}$. It also defines a set $\mathscr{L}_a \subset \mathscr{L}$ of request locations that need to be covered by the selected sensor-types of the mobile devices in those locations. The formulation here assumes that each location will have at least one mobile device present. Partial covering or no covering of the location (when no device is present) is not addressed and will be covered as part of future work. Fig. 3.2b shows multiple applications requesting for different request locations in the grid and Fig. 3.2c shows how mobile devices in each request location cover the area requested by the application in time interval k. Each application a also specifies a minimum sampling rate S_M^a for each sensor-type $M \in \mathscr{M}_a$ and the sensed data needs to be reported to the application at a constant time period termed as reporting time period. This helps in creating a set of reporting time-points for each sensor-type M needed by the application a which is denoted by the set τ_M^a. Thus $\bigcup_{a \in \mathscr{A}, M \in \mathscr{M}_a} (k-1)\Delta t < t \in \tau_M^a < k\Delta t$ indicates the applications and the sensor-types for which sensed data is requested and has to be reported in the time interval k.

It should be noted that if time interval k contains a reporting time point of sensor-type $M \in \mathscr{M}_a$ for any application $a \in \mathscr{A}$, then the information collected but not reported is available to process and report.

3.3.2 Problem Statement

Given the set of mobile devices and their embedded sensors, all the sensing requirements of the application must be met over the total duration of $\mathscr{T}$ seconds. It is further assumed that the variance of sensed data during an interval does not affect the accuracy required by the application. This is formulated as an optimization problem for the selection of sensors on particular devices to activate, the sampling rate they should use and the quantity of sensed data they should offload to the cloud for further processing.

3.3.2.1 Decision Variables

A mobile device is regarded as being active if it senses data for one or more sensor types during the Δt time interval. The decision variable $y_n^k \in \{0,1\}$ indicates if device n is active during time interval k. Furthermore, $y_{nm}^k \in \{0,1\}$ indicates whether device n is active for the sensor-type m requested by the application during time interval k. The relation between these variables is defined in Eq. 3.1.

$$\begin{gathered}\forall k \in [1, \mathscr{T}/\Delta t], \forall n \in \mathscr{N}: \\ \sum_{m \in \mathscr{M}_n} y_{nm}^k \geq 1 \Longrightarrow y_n^k = 1\end{gathered} \tag{3.1}$$

Multiple applications may rely on one sensor to conduct sampling during time interval k. For every sensor-type $m \in \mathscr{M}_n$ on device n, r_{nm}^k is defined as the decision variable to indicate the sampling rate during time interval k determined by the highest rate required by all applications using it. The mobile device must be able to provide this sampling rate for the sensor-type, as defined in Eq. 3.2:

$$\begin{aligned}&\forall k \in [1, \mathscr{T}/\Delta t], \forall n \in \mathscr{N}, \forall m \in \mathscr{M}_n : r_{nm}^k \in \mathscr{S}_m \\ &r_{nm}^k >= 1 \Longrightarrow y_{nm}^k = 1\end{aligned} \tag{3.2}$$

Additionally, processing of the sensed data can be done locally or offloaded to data centres for applications requiring different accuracy levels, which means some partial sensed data processing might be done locally to reduce the network bandwidth consumption. The decision variable o_n^k is defined to decide the percentage of the data that should be offloaded directly for device n in time interval k. The accumulated volume of sensed data for the

Table 3.1 Notation used for Problem Formulation in Chapter 3.

Notation	Description
$\mathscr{T}$	Planning horizon
t	Time variable
Δt	Time Interval for aggregation
k	Index of time interval
$\mathscr{L}$	Set of Detection Locations in the sensing area
$\mathscr{L}_i$	Set of 2D coordinates for the ith Location
$\mathscr{N}$	Set of mobile devices
n	Index of mobile device
$(x_n(k), y_n(k))$	Coordinates of the mobile device in time interval k
$\mathscr{N}_i^k$	Set of mobile devices in request location $\mathscr{L}_i$ for time interval k
$\mathscr{H}$	Set of all sensor-types for available mobile devices
$\mathscr{M}_n$	Set of sensor-types on device n
m	Index of sensor-types in the mobile device
$\mathscr{S}_m$	Set of sampling rates of sensor-type m
v_m^n	Volume accumulated by sensor-type m in the mobile device n
V_n^k	Total Volume accumulated by mobile device n in time interval k
$\mathscr{A}$	Set of applications
a	Index of applications
$\mathscr{M}_a$	Set of sensor-types requested by application a
M	Index of sensor-types requested by application a
$\mathscr{L}_a$	Set of request locations of application a constant in time
S_M^a	Minimum sampling rate of sensor-type m for application a
τ_M^a	Set of reporting time points of sensor-type M for application a
y_{nm}^k	0-1 variable to indicate if a sensor m is active in mobile device n during time interval k
y_n^k	0-1 variable to indicate if a device is activated during time interval k
r_{nm}^k	Integer reflecting sampling rate of sensor-type m on device n during time interval k
r_{nm}^{ak}	Integer reflecting minimum sampling rate of sensor-type m on device n during time interval k for application a
o_n^k	Percentage of data offloaded directly for device n during time interval k

Notation	Description
E_n^k	Energy consumption related to sensing for device n during time interval k
F_n^k	Local information processing cost in terms of energy consumption
G_n^k	Information offload cost in terms of energy consumption
p	Percentage of information volume after processing
ζ	Minimum percentage of battery level
B_n	Full batter level of device n
b_n^t	Battery level of device n at the start of time interval k
γ, δ	Normalization parameters to balance battery consumption, active mobile devices and volume offloaded into cloud servers

device n is represented by $V_n^k = \sum_{m \in \mathscr{M}_n} v_m^n . y_{nm}^k$, and it does not matter if this sensed data is used by one application or multiple applications. It is further assumed that computation and offloading of the sensed data can be done simultaneously. A dynamic decision must be made to keep information processing locally or offloaded to cloud.
Using the decision variable defined above, the volume of sensed data directly offloaded is given by $o_n^k \cdot V_n^k$ and then $(1 - o_n^k) \cdot V_n^k$ is the volume which will be processed locally. After the preprocessing and aggregation, the volume of $(1 - o_n^k) \cdot V_n^k$ is reduced to $p(1 - o_n^k) \cdot V_n^k$ and simultaneously offloaded.

3.3.2.2 Constraints

This subsection defines the constraints relating to the sensor coverage area, accuracy required by the application and minimum device battery level as follows.
1. Coverage Constraint:
Each application a specifies request locations $\mathscr{L}_a$ to be covered for time interval $k \in [1, \mathscr{T}/\Delta t]$. Eq. 3.3 ensures that these areas are each covered by one or more sensors on one or more active mobile devices.

$$\begin{aligned} &\forall k \in [1, \mathscr{T}/\Delta t], \forall a \in \mathscr{A}, \forall \mathscr{L}_i \in \mathscr{L}_a, \forall M \in \mathscr{M}_a, \forall n \in \mathscr{N}_i^k : \\ &\mathscr{N}_i^k \neq \emptyset \\ &\textstyle\sum_{\mathscr{N}_i^k} y_n^k \geq 1 \end{aligned} \tag{3.3}$$

2. Accuracy Constraint:
The accuracy constraint for each application must be met by the sensor-types on the active mobile devices. Coordinated sampling by multiple sensors at lower sampling rates to achieve the required accuracy is not considered. The decision variable r_{nm}^{ak} is used to indicate the sensor sampling rate during time interval k of sensor-type m on device n when only satisfying the requirement of application a. The value of r_{nm}^{ak} will be greater than zero only if application a has a reporting time-point in the time interval k for sensor-type m. Eq. 3.4 defines the accuracy constraint.

$$\begin{aligned}
&\forall k \in [1, \mathscr{T}/\Delta t], \forall a \in \mathscr{A}, \forall n \in \mathscr{N}, \forall m \in \{\mathscr{M}_n \cap \mathscr{M}_a\}: \\
&r_{nm}^{ak} > 0 \Longrightarrow \exists \tau \in \tau_m^a : (k-1)\Delta t < \tau < k\Delta t \\
&r_{nm}^{ak} > 0 \Longrightarrow r_{nm}^{ak} >= S_m^a \\
&r_{nm}^{ak} <= r_{nm}^{k}
\end{aligned} \tag{3.4}$$

3. Battery Life Constraint:
Energy costs may be incurred by a device n during time interval k related to the monitoring, sensing, processing and offloading activities. The battery consumption level of device n for sensing activities during time interval k, is denoted by E_n^k. This is defined based on a mapping between the values of all decision variables r_{nm}^k, which indicates the sampling rates of sensor-type m on device n. Energy consumed for local processing of sensor data is denoted by F_n^k and is a function of $((1-o_n^k)\cdot V_n^k)$. It is assumed that this energy for data processing is only expended during the last period of the time interval. The battery consumed to offload data of volume $o_n^k \cdot V_n^k + p \cdot (1-o_n^k)\cdot V_n^k$ to the cloud server is denoted by G_n^k. If the battery level of device n at the start of time interval k, denoted b_n^k, is below ζ of the full battery level B_n, it is pre-empted due to possible power off during the sensing or reporting activities. This restriction is imposed on all mobile devices to ensure battery availability for sensing, local data pre-processing and information offload into the cloud and defined in Eq. 3.5

$$\begin{aligned}
&\forall n \in \mathscr{N}, \forall k \in [0, \mathscr{T}/\Delta t]: \\
&b_n^k >= \zeta B_n \\
&b_n^k - E_n^k - F_n^k - G_n^k = b_n^{k+1}
\end{aligned} \tag{3.5}$$

3.3.2.3 Objective Function

During each time interval k, a deployment has a set of possible mobile devices with embedded sensors that are activated to provide information at different sampling rates for applications. The objective is thus threefold:

- Minimize energy consumption due to sensing, local data processing and information offload into the mobile cloud. This is reflected in the first term which is a summation of the energy losses incurred in the process;
- Minimize number of active mobile devices in the sensing environment. This is represented by the second term which is the summation of the active mobile devices in the system to which a weight of γ is allotted;
- Minimize the volume of data available to be offloaded into the cloud. This is represented in the third term which is allotted a weight of δ.

These can be expressed formally in Eq. 3.6.

$$\text{minimize} \quad \sum_{k=1}^{\mathscr{T}/\Delta t} \left(\sum_{n \in \mathscr{N}} (E_n^k + F_n^k + G_n^k) + \gamma \sum_{n \in \mathscr{N}} y_n^k + \delta \sum_{n \in \mathscr{N}: y_n^k = 1} (o_n^k \cdot V_n^k + p \cdot (1 - o_n^k) \cdot V_n^k) \right) \tag{3.6}$$

3.4 Algorithm Design

Because of the non-polynomial complexity of the problem formulation presented above, it might take several hours or even days (using commercial optimization software without parameter configurations), to produce a solution even for a relatively small instance. Therefore, a fast, even if sub-optimal, approach is required for efficient selections of the sensors for satisfying application requirements. This section introduces two algorithms, namely Algorithm No-Aggregation and Algorithm Info-Aggregation that run in the framework to create a schedule of activated sensors embedded in mobile devices for offloading sensed data to multiple applications.

The first, Algorithm No-Aggregation is described in §3.4.1 and is a simple algorithm used as a baseline for comparison. It seeks to minimise the energy expended by mobile devices without being aware of the ability to aggregate sensed data, by serving individual datums to multiple applications. The latter, Algorithm Info-Aggregation is described in §3.4.2 and handles aggregation by initiating a trade-off between energy-efficiency and the volume of data offloaded by active mobile devices. This algorithm is based on the widely studied pattern mining technique (Han et al., 2004, 2006) as it enables unsupervised learning and allows patterns to be found for all kinds of data and large-datasets. The adoption of a revised version of the Frequent Pattern Growth algorithm (Han et al., 2007a) helps in supporting a divide-and-conquer strategy for producing frequent patterns. Both these algorithms utilise the Truncated Levy Walk mobility model for prediction of the mobile device coordinates within the physical sensing area of interest to the applications. This is adopted based on recent studies of human mobility (Karamshuk et al., 2011) that show a similarity between such Levy walks and human mobility. Rhee et al. (2011) have presented a truncated Levy walk model which provides a simple and realistic model for human mobility; it is this variant that has been presented in the simulation model.

For specifying both the algorithms, certain terminology is used that helps in explaining the flow of the algorithms. The first term defined for this purpose is the *Application-Sensor* pair which is mathematically represented as $< a, M >$. Since every application a requests for sensed information from specific sensor-types $M \in \mathscr{M}_a$, this term helps in uniquely identifying $\mathscr{L}_a$ and the set of reporting time-points τ_M^a needed. It can be computed that the total number of Application-Sensor pairs will be $\sum_{a \in \mathscr{A}} |M_a|$. Additionally, multiple mobile devices would have the same sensor-type embedded in their system. This means that multiple devices are capable of providing the sensed information for a particular request location and multiple combinations of mobile devices can be created that will cover all the location constraints as specified by the Application-Sensor pair, when activated concurrently. One such set of mobile devices is defined as a *NodeSet* and used by the algorithms to satisfy Application-Sensor pairs. To calculate the energy expended by the NodeSet, a summation of the individual energy consumed by each mobile device in the NodeSet is used. This terminology has also been referred in Chapter 4 of this dissertation.

The flow of both algorithms is as follows. First, for every Δt interval, the request location covered by each mobile device is predicted. For this, it is assumed that the starting positional coordinates can be retrieved by the framework using GPS, WiFi Positioning or some similar technology. The function *predictLocOfMobileDevice(n, t.start, t.end)* is then applied to predict the request location covered by each mobile device for the entire duration of the time interval. This is determined by adopting a look-ahead mechanism and using an instance of the mobility model, which helps in deciding the aggregation that can be made for the request location. For any application $a \in \mathscr{A}$ and sensor-type $M \in \mathscr{M}_a$, if $\forall k \in \mathscr{T}/\Delta t, \exists \tau \in \tau_M^a : (k-1)\Delta t < \tau < k\Delta t$, then that Application-Sensor pair is present in the time interval k. This is determined by the framework that sorts and maps all reporting time-points. Using this formulation, the pairs are identified and stored in the data structure AppSensorPairs[]. For the request locations of a Application-Sensor pairs in the AppSensorPairs[] collection, the valid NodeSets are ascertained such that each location is covered by one or more mobile devices. A NodeSet is considered valid only when the movements of each mobile devices in the NodeSet enables it to cover the request location for

the entire time interval k. It is assumed that variance in the movement of the mobile device within the interval does not affect the sensing readings, if it is still in a position to cover the location. Following this, one NodeSet is selected for each Application-Sensor pair and the mobile devices are activated to sense data. Finally, the energy consumed by each mobile device for the time interval k is calculated using function *calcEnergyConsumption(k)*.
To calculate the percentage of data offloaded for processing in the cloud, each active mobile device acts in isolation. It calculates the cost for local processing and the cost of offloading data directly into the cloud to select an optimal offload percentage, denoted in the formulation as o_n^k, that reduces the sum of both involved costs. This percentage is calculated using a simple local search heuristic approach, with the assumption that statically processed results of historical offloading costs are available in the collaborative sensing framework. These results are used to understand the optimal range of offload percentage for various network states, which is assumed to be related to the number of other active mobile devices in the system. On the other hand, local processing costs of the data are defined to be a simple weighted function of the amount of data to be offloaded and the available battery in the mobile device.
For $o_n^k = 0$ indicating that none of the data is offloaded by the device for cloud processing and $o_n^k = 100$ indicating that the entire sensed data V_n^k is offloaded by mobile device n, the following assumptions are considered:

1. For $x > y$, if the cost involved in offloading $o_n^k = x$ percent is worse than $o_n^k = y$ percent, then $\forall z : z > x$, offloading $o_n^k = z$ percent is worse than offloading $o_n^k = y$ percent.

2. For $x > y$, if cost involved in offloading $o_n^k = x$ percent is better than offloading $o_n^k = y$, then it is possible that $\exists z > x$ where offloading $o_n^k = z$ is better than offloading $o_n^k = y$ percent.

3. For $x < y$, if cost involved in offloading $o_n^k = x$ percent is worse than offloading $o_n^k = y$ percent, then $\forall z : z < x$, offloading $o_n^k = z$ is worse than offloading $o_n^k = y$.

4. For $x < y$, if cost involved in offloading $o_n^k = x$ percent is better than offloading $o_n^k = y$ percent, then it is possible that $\exists z < x$ where offloading $o_n^k = z$ is better than offloading $o_n^k = y$ percent.

Using these assumption, function *calcPercentageToOffload(k)* recursively determines the offloading percentage that minimizes the sum of the local processing cost and the offloading

Algorithm 3.1 Algorithm No-Aggregation

for k=1 to $\mathscr{T}/\Delta t$ **do**
    $t.start$=$(k-1)\Delta t$
    $t.end$=$k\Delta t$
    for n =1 to $\mathscr{N}$ **do**
        predictLocOfMobileDevice(n, $t.start$, $t.end$)
    end for
    AppSensorPairs[]=getAllAppSensorPairs(k)
    for all AppSensorPairs[] **do**
        loc l[]=getLocs(AppSensorPairs[],k)
        NodeSets=calcNodeSetForLoc(l[],k)
        for all NodeSets **do**
            getEnergyConsumed(NodeSet,k)
        end for
        Activate NodeSet with minimum battery for each Application-Sensor Pair.
    end for
    Run CALCENERGYCONSUMPTION(k)
    Run CALCPERCENTAGETOOFFLOAD(k)
end for

cost with a procedure similar to the binary search technique. This function starts by taking $o_n^k = P$ as the starting point, where P is randomly taken from the analysed optimal range of offload percentages. Then, two threads are run for randomly-selected neighbours of P, namely $T1 > P$ and $T2 < P$. Next, the cost involved in offloading $T1\%$ and $T2\%$ of the sensed data and local processing of the remaining percentage is calculated. The thread which increases the costs involved is aborted, and the control passes to the thread with the lower cost, thereby reducing the search space. Thus, this recursive technique helps determine the optimal percentage of data that should be offloaded. Future work of this dissertation will focus on improving this methodology and incorporating other techniques for optimal selection of the percentage of data to be offloaded directly into the cloud.

The following subsections explain the workings of the two algorithms, in detail and an analysis of their complexity is also presented in §3.4.3.

3.4.1 Algorithm No-Aggregation

This subsection defines the algorithm No-Aggregation (Algorithm 3.1) whose only objective is to reduce energy-consumption. Each application is treated independently, so that opportunities to aggregate information from sensors and serve these to multiple applications are not identified. This may lead to unnecessary duplicate transmissions of data to the applications and consequently overall energy depletion.

Once the NodeSets for each Application-Sensor pair have been determined, the algorithm calls function *getEnergyConsumed()* in line 12 to calculate the energy consumed by the NodeSet. For each Application-Sensor pair, the algorithm selects the NodeSet that expends minimum energy. The devices present in that NodeSet are activated for the time interval.

3.4.2 Algorithm Info-Aggregation

The objective of Algorithm Info-Aggregation(Algorithm 3.2), is threefold as presented in §3.3. It focuses on minimizing the energy consumed during sensing, offloading and local processing along with reducing the number of active devices and the volume of offloaded sensed data. This is made possible by multitasking the capability of a single mobile device to satisfy the constraints of more than one Application-Sensor pair. In this algorithm, the selection of the NodeSet is made only after calculation of the frequently occurring subset of mobile devices amongst all NodeSets present in the time interval. For this, it is important to define and calculate the most frequently occurring subset of mobile devices.

Assuming all subsets of mobile devices, with sizes ranging from one to sizes equal to the maximum number of mobile devices that are present in all NodeSets collectively, are considered in this calculation. Then the subset which occurs the most in the NodeSets for all Application-Sensor pairs will be defined as the most frequently occurring subset. However, counting the frequency of each subset is highly computationally intensive. By applying data mining techniques(Han et al., 2006) for frequent pattern mining, this computation can be reduced. The revised FP-Growth algorithm is selected for this purpose(Han et al., 2004), which is deployed as Functions 3.3 and Function 3.4 and explained in this section. This is preferred over other data mining algorithms like Apriori (Han et al., 2006) to reduce the complexity of searching through all possible combinations of the mobile devices to find frequent occurring subsets. The compact tree structure helps in saving memory space as well. The principle of this algorithm is that if a subset of $(w+1)$ mobile devices is frequent, then the subsets of w mobile devices derived from this set will also be frequent. To prune out subsets, it is assumed that any subset that occurs less frequently than a predefined proportion of NodeSets will not affect the traffic during run-time. This proportion of the number of times a subset occurs in a time interval is termed as the *support* of the set and a minimum value of the support called *minimum support* is required by any subset to be of benefit during

Algorithm 3.2 Algorithm Info-Aggregation

1: **for** k=1 to $\mathscr{T}/\Delta t$ **do**
2:     $t.start$=$(k-1)\Delta t$
3:     $t.end$=$k\Delta t$
4:     **for** n =1 to $\mathscr{N}$ **do**
5:         predictLocOfMobileDevice(n, $t.start$, $t.end$)
6:     **end for**
7:     AppSensorPairs[]=getAllAppSensorPairs(k)
8:     **for all** AppSensorPairs[] **do**
9:         loc l[]=getLocs(AppSensorPairs[i],k)
10:         NodeSets=calcNodeSetForLoc(l[],k)
11:     **end for**
12:     Tree T=FP-TREE(NodeSets)
13:     FP P=FP-GROWTH($T.root$, $null$)
14:     Sort P according to size and support of subsets
15:     haveFrequentSet = false;
16:     **for all** Set p in P **do**
17:         **if** hasAvailableBattery(p)==true **then**
18:             haveFrequentSet=true;
19:         **end if**
20:     **end for**
21:     **if** haveFrequentSet==true **then**
22:         COVERAPPSENSOR(p, AppSensorPairs[])
23:     **else**
24:         NO-AGGREGATION
25:     **end if**
26:     Run CALCENERGYCONSUMPTION(k)
27:     Run CALCPERCENTAGETOOFFLOAD(k)
28: **end for**
29: **function** COVERAPPSENSOR(p, AppSensorPairs[])
30:     **for all** AppSensorPairs[] **do**
31:         getNodeSetsWithMinDistance(AppSensorPairs[], p)
32:         Sort by energy consumed
33:         Activate NodeSet with min-Distance and min-energy consumption from p
34:     **end for**
35: **end function**

computation and termed frequent. Thus the subset must occur at least in the proportion specified by the minimum support.

The proposed algorithm Info-Aggregation (Algorithm 3.2) runs in the following manner. Once all NodeSets have been determined, the function Function 3.3 is used in line 12 for creating the frequent pattern (FP) tree also known as the FP-Tree. For this, first the

Function 3.3 Function FP-Tree

```
1: function FP-TREE(NodeSets)
2:     for all n ∈ NodeSets do
3:         getFrequentNodes()
4:     end for
5:     Order each device in NodeSets in terms of frequency of individual device and add to
   a header list
6:     T = Tree with Empty Root
7:     for all n ∈ NodeSets do
8:         INSERTINTOTREE(T.root, n)
9:     end for
10:    return T
11: end function
12: function INSERTINTOTREE(root, n)
13:    if n == ∅ then return ;
14:    else
15:        prefixNode = n.firstNode
16:        if ChildOfRoot.equals(prefixNode) then
17:            ChildOfRoot.freq += 1
18:            Connect Node to HeaderList
19:            return INSERTINTOTREE(ChildOfRoot, n − prefixNode)
20:        else
21:            prefixNode = new ChildOfRoot
22:            prefixNode.freq = 0
23:            Connect Node to HeaderList
24:            return INSERTINTOTREE(prefixNode, n − prefixNode);
25:        end if
26:    end if
27: end function
```

frequency of the individual (all subsets of size one) that occur across all these NodeSets is counted by using the function $getFrequentNodes()$ in line 3 of Function 3.3. This helps in ordering the NodeSets in decreasing order of support of the individual mobile devices of the NodeSet. The FP-Tree is created by defining a root device that points to null. Next, each NodeSet is inserted into the tree using the recursive procedure outlined from line 12 to 26 of Function 3.3 which aims at creating a tree where different NodeSets sharing a common prefix are attached to a single child of the root device of the tree. The halting condition of the recursion occurs when the NodeSet to be added is an empty set. If this condition is false, the first mobile device in that NodeSet is assigned to be the prefix. The children of the root device are traversed to match with this prefix. If any child of the root device already contains

Function 3.4 Function FP-Growth

```
1: function FP-GROWTH(T.root, α)
2:     if T contains a path P then
3:         for all combination β ∈ P do
4:             sample = β ∪ α
5:             sample.freq = count of β
6:             if sample.freq ≥ minSupport then
7:                 add sample to P;
8:             else
9:                 end;
10:            end if
11:        end for
12:    else
13:        for all aI ∈ Header List do
14:            β = aI ∪ α
15:            freq of β = count of aI
16:            Construct conditional T∗ for β
17:            if T∗ is non-empty then
18:                FP-GROWTH(T∗, β)
19:            else
20:                continue
21:            end if
22:        end for
23:    end if
24: end function
```

this mobile device, the rest of the NodeSet is added as children to this branch of the tree. Otherwise, a new branch is created for the NodeSet and devices following the prefix are created and linked accordingly. In general, when considering the branch to be added, the count of each device along a common prefix is incremented so that the devices have information about their frequency. To facilitate tree traversal, a header table is built of the individual mobile devices so that each device points to its occurrences in the tree via a chain of device-links.

After the creation of the FP-Tree, the recursive function 3.4 mines the FP-Tree for frequently occurring subsets in a breadth-first fashion. This means that the subsets where the particular mobile device acts like a suffix leaf are examined first, thereby improving selectivity. For each of its occurrences, the device-links are used to find the different prefix paths (from the ancestors in the tree) and their frequencies, which is equal to the count of the leaf device. In line 16 of Function 3.4, a conditional tree is built using the prefix paths of the current leaf device or suffix while excluding the suffix device. This conditional tree only contains the count as present in the prefix paths. If the tree is non-empty, it is mined further. If the tree contains a single child device then for this path, the various subsets are found and included in the frequent pattern collection, only if their count is more than the minimum support. The FP-Growth method thus transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix.

These frequent patterns (FP) obtained after mining, are stored in the variable P in line 13 of Algorithm 3.2. The subsets are sorted according to the size and support of the frequent pattern in the decreasing order. The next task is to select a frequent set of mobile devices which forms the *base subset* to be used in selecting one NodeSet for all Application-Sensor pairs. This base subset should have a high support, enough battery and be of a good size so that most of the mobile devices in that subset match the NodeSet that is selected to cover the Application-Sensor Pair. To enable this, the battery available in the mobile devices of each subset is iteratively checked in decreasing order of size and support. In line 21, if a frequent set meeting the battery constraints is present, it is selected as the base subset p to be used to cover the Application-Sensor Pairs. The process of selecting one NodeSet by using the base subset is explained from line 29 to 35 of the procedure *coverAppSensor*. For each pair, the NodeSet that satisfies the Application-Sensor pair but includes most of the active mobile devices in the base subset is thus determined and selected.

3.4.3 Complexity Analysis

The worst case running time for both these algorithms, when each application requests for sensed data from all available sensors, is analysed and presented in this sub-section.

The total of number of Application-Sensor pairs can be calculated by $\sum_{a\in\mathscr{A}} |M_a| = |\mathscr{A}| \cdot |\mathscr{H}|$, where $|\mathscr{H}|$ is the maximum number of sensor-types that can be embedded in a mobile device. Assuming that the maximum number of request locations covered by all application is given $\forall a, |\mathscr{L}_a| = l$ with a maximum of $d <= |\mathscr{N}|$ mobile devices for the time interval k, the total of number of NodeSets for each Application-Sensor pair can be calculated to be d^l.

For Algorithm No-Aggregation (Algorithm 3.1), the complexity of selecting one NodeSet for each Application-Sensor pair, can thus be represented as $O(|\mathscr{A}| \cdot |\mathscr{H}| \cdot d^l)$. However, to accurately represent the complexity of Algorithm Info-Aggregation (Algorithm 3.2), the worst-case time complexity of the FP-Growth algorithm needs to be calculated. This is dependent on the complexity of searching for frequent patterns in the FP-Tree and is proportional to the number of unique elements, $d \cdot l$ present in the header table created in

Function 3.3 as well as the depth of the tree. For the worst case analysis, this tree will be an unbalanced tree and its depth will be upper-bounded by $d \cdot l$. Thus the complexity of traversing through all paths is $O(d^2 \cdot l^2)$. In consequence, using the FP-Tree ensures that the complexity of this algorithm is much less than searching through all possible combinations which is given as 2^{dl}.

3.5 Evaluation

This section evaluates the performance of Algorithm Info-Aggregation (Algorithm 3.2) in comparison to Algorithm No-Aggregation (Algorithm 3.1) presented in §3.4. To achieve this, the number of applications $|\mathscr{A}|$, mobile devices $|\mathscr{N}|$, sensor-types $|\mathscr{H}|$ and requirements of the applications are varied.
The main difference between the algorithms is the ability of Algorithm Info-Aggregation to identify the potential of serving sensed data from one mobile device to multiple applications, thereby reducing redundancy of data offloaded into the cloud for processing. The results suggest that Algorithm Info-Aggregation effectively reduces the number of active mobile devices and volume available for offloading from the sensing environment whilst saving energy expended in the process.
The section is organized as follows. It begins by detailing the simulation model, illustrating experimental results and concludes by presenting the findings.

3.5.1 Simulation Model

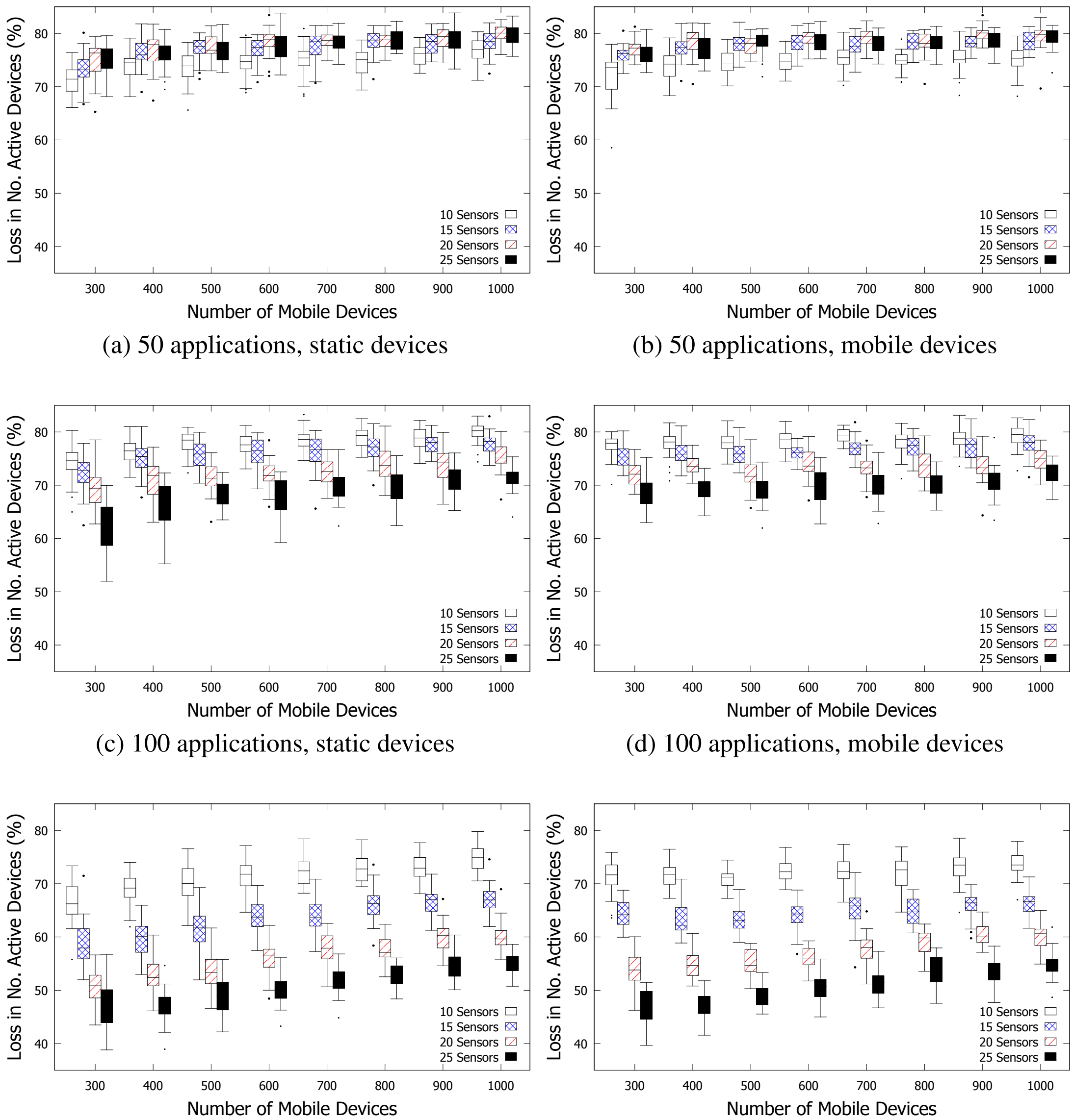
The physical sensing area is modelled using a grid (100m $\times$ 100m) subdivided into request locations (10m $\times$ 10m), considering the dimensions of an average house in Ireland/United Kingdom[1]. This represents the simulation study area and also bounds the trajectory of each mobile device. To emulate device movements, the Truncated Levy-Walk mobility model (Rhee et al., 2011) is selected which is represented by the tuple (l, θ, t_f, t_p). In this definition, l is the flight length randomly picked up from a Levy distribution with coefficient $\alpha = 1.5$, θ is the angle of flight which follows a uniform distribution, t_f is the flight time, and t_p is the pause time which is Levy distributed with coefficient $\beta = 0.5$. The truncation factors are defined as 100m and 1000s respectively for the flight length and pause times. The real path taken by the mobile device during the simulation study is defined using one instance of this model while another instance is used to predict the path taken by the mobile device

[1] How Big is a House? http://shrinkthatfootprint.com/how-big-is-a-house

for the next time-interval. These time-intervals are defined as $\Delta t = 2$ minutes to represent time progression within the simulation over a total simulated duration of $\mathscr{T} = 480$ minutes. Each application $a \in \mathscr{A}$ specifies the specific sensor-types for which data is requested from different locations within the grid at a predefined reporting rate. It is assumed that an application requests for a maximum of four locations to be covered by a requested-sensor type. These sensor-types are uniformly distributed on the mobile devices that are modelled to be spread across the physical sensing area. The device is assigned a full battery, equivalent to 5.45Wh (Apple) at the beginning of every simulation run. The decrease in the battery level of the mobile device is associated with the energy consumed for general usage of the mobile device, for operation of the sensors, and for transmission of offloaded sensed data. Every sensor-type that is accessed from this mobile-device contributes to this loss of battery which is specific to that sensor-type. Sensirion offers environmental sensors for mobile devices with energy consumption as low as 2μW (Sensirion, a,c) while LittleRock (Priyantha et al., 2011) presents a table on power consumed by different sensor-types. These values are consulted for randomly assigning the energy expended by the sensor-type within the range 0.002mW to 2.24mW (Priyantha et al., 2011) to cover different sensor-types. The percentage of decrease for general usage and user interaction of the mobile device is randomly selected. Another variable associated with the sensor-type is the volume of sensed data that it collects over time, this is randomly selected between 8 bits/second (Sensirion, b) to 50 bits/second. This data is then fully or partially offloaded to the cloud for further processing. For energy transmission calculations, the mobile device loses 0.05W (Balasubramanian et al., 2009) to maintain WiFi connections. When transferring sensed data, the transmission energy model as presented by Friedman et al. (2013) across WiFi (ad-hoc and with access-points) is used to send data using TCP/UDP. These communication protocols and the WiFi networks (ad-hoc or access-points) used for this transmission are randomly selected in an endeavour to cover different transmission channels.

3.5.2 Results and Analysis

A single experiment of this study is defined by a fixed number of applications, number of mobile devices, sensor-types in the sensing area and the mobility pattern of the mobile devices. It is simulated thirty times by using different random number generator seeds. The number of applications are varied between 50, 100 and 200, with total sensor-types being varied between 10, 15, 20 and 25, and the number of mobile devices present in the environment to sense, collect, pre-process and offload this sensor data selected in the range



(a) 50 applications, static devices

(b) 50 applications, mobile devices

(c) 100 applications, static devices

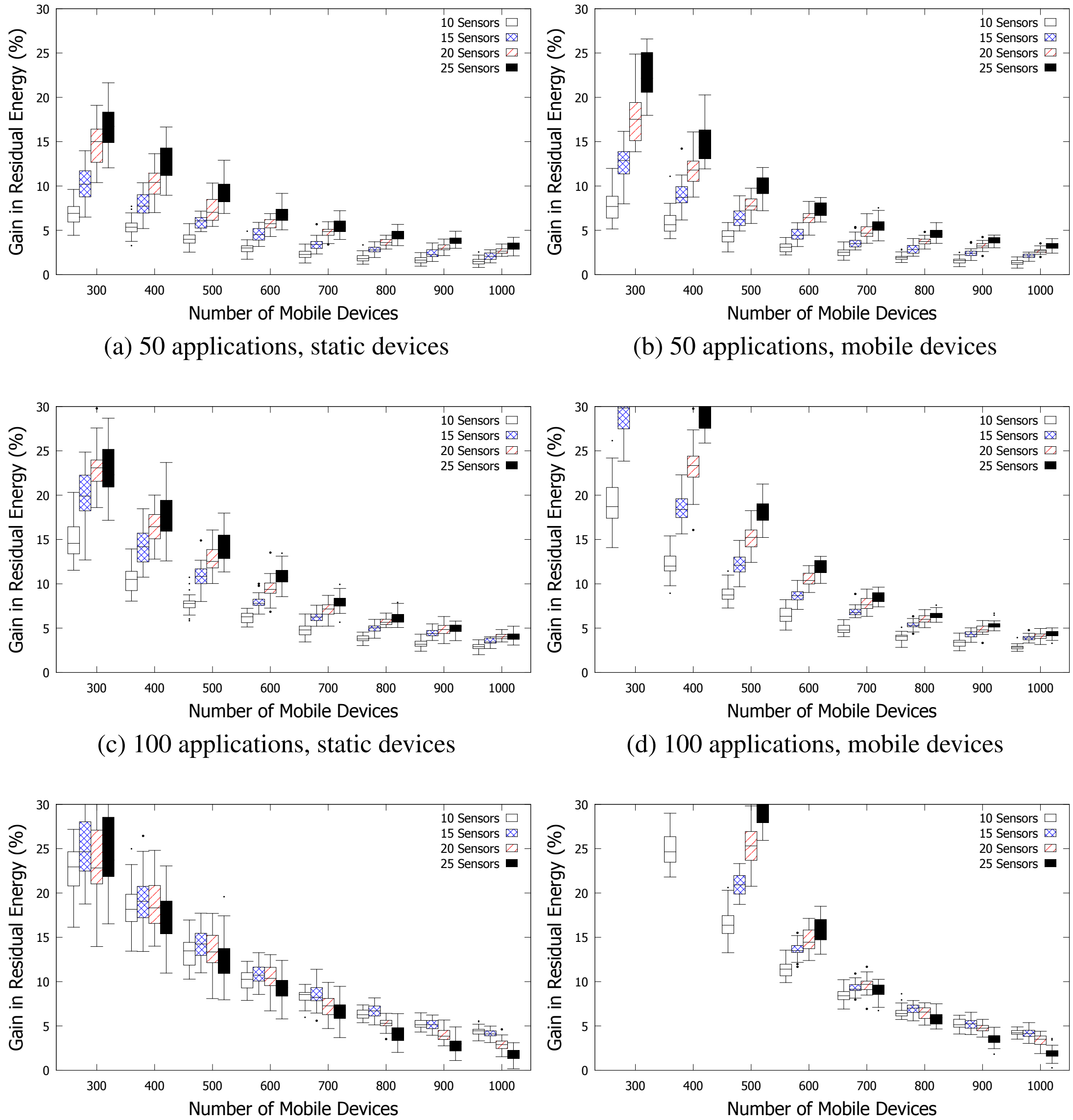
(d) 100 applications, mobile devices

(e) 200 applications, static devices

(f) 200 applications, mobile devices

Fig. 3.3 Comparative mean percentage difference in the number of active mobile devices offloading sensed data to the cloud between the two algorithms, Algorithm No-Aggregation and Algorithm Info-Aggregation vs. the total number of mobile devices in the sensing environment for 50, 100 and 200 applications. It can be seen that in all cases, Algorithm Info-Aggregation requires the activation of significantly fewer mobile devices, especially as the number of devices in the environment changes.

of 300 to 1,000. Two mobility patterns of the mobile devices are considered. The first model defines static mobile device while the latter incorporates the Truncated Levy Walk mobility model, defined above. This is done to ensure that the advantage of aggregation of sensed



(a) 50 applications, static devices

(b) 50 applications, mobile devices

(c) 100 applications, static devices

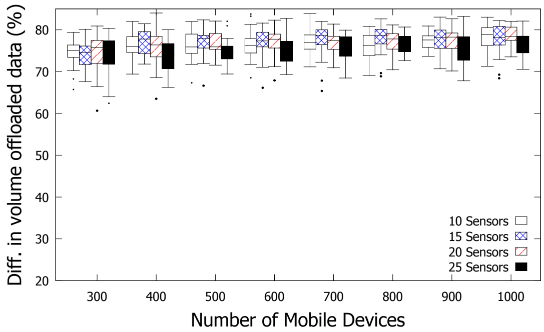
(d) 100 applications, mobile devices

(e) 200 applications, static devices

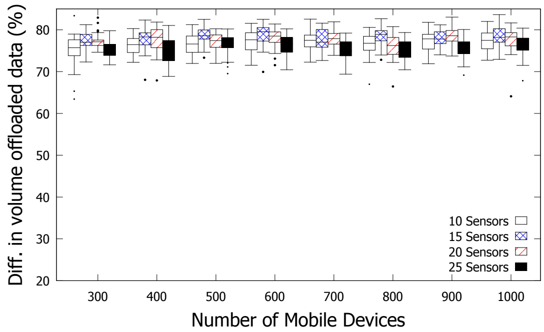
(f) 200 applications, mobile devices

Fig. 3.4 Comparative percentage difference in the cumulative residual energy held in mobile device batteries between the two algorithms, Algorithm No-Aggregation and Algorithm Info-Aggregation vs. the total number of mobile devices in the sensing environment for 50, 100 and 200 applications. It can be seen that Algorithm Info-Aggregation results in a lower level of cumulative energy use by mobile devices, due both to the sensors in fewer devices being activated and to a lower number of messages with sensed data being transmitted due to the use of aggregation.

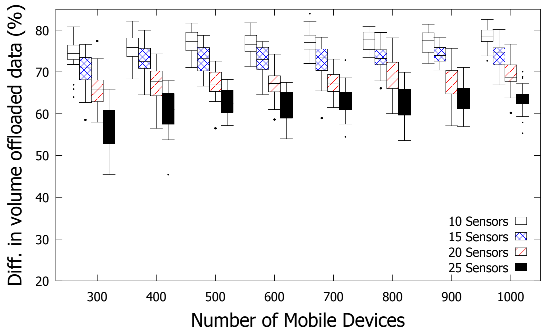
data from minimally active mobile devices for multiple applications can be considered for both cases.



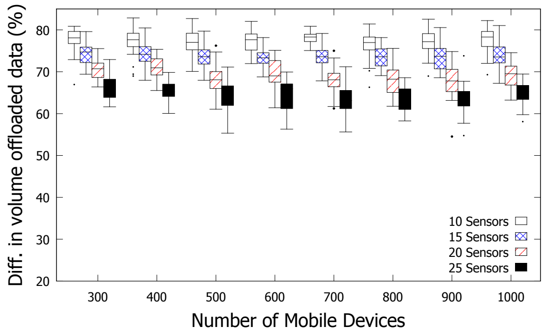
(a) 50 applications, static devices



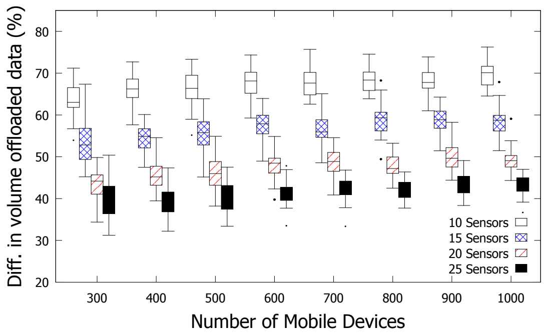
(b) 50 applications, mobile devices



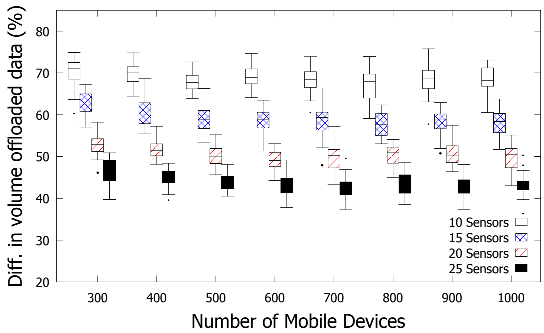
(c) 100 applications, static devices



(d) 100 applications, mobile devices



(e) 200 applications, static devices



(f) 200 applications, mobile devices

Fig. 3.5 Comparative mean percentage difference in the volume of sensed data offloaded to the cloud between the two algorithms, Algorithm No-Aggregation and Algorithm Info-Aggregation vs. total number of mobile devices in the sensing environment over three cases using 50, 100 and 200 applications. The use of aggregation means that the Algorithm Info-Aggregation offloads a lower volume of data than does Algorithm No-Aggregation.

For each experiment, the performance of Algorithm Info-Aggregation is compared with Algorithm No-Aggregation in terms of the gain in cumulative residual energy values of all mobile devices, the difference in volume of data available for offloading and the difference

in the mean number of active mobile devices in the sensing area during the planning horizon. The percentage difference between the values is recorded for the two algorithms.
Fig. 3.3 shows the mean reduction in the number of activated mobile device for Algorithm Info-Aggregation in comparison to Algorithm No-Aggregation for both static and dynamic mobility patterns. These results show that the number of active devices can be reduced by more than 40% when sensed data from mobile devices is shared between multiple interested applications. Also, the percentage difference increases as the total number of mobile devices in the sensing environment increase. This shows a clear benefit to aggregating sensor data for multiple interested applications.
Fig. 3.4 shows the mean increase in cumulative residual energy stored in mobile device batteries at the end of the simulated time interval for Algorithm Info-Aggregation in comparison to Algorithm No-Aggregation for both static and dynamic mobility patterns. This increase is due to the decreased number of activated mobile devices needed by the Algorithm Info-Aggregation to serve the required sensed data to the applications, and to the impact of fewer message transmissions due to the use of aggregation. As the number of applications increases this effect is amplified, as the same sensor data is requested by multiple applications. It can also be observed that the cumulative energy gain decreases as the number of mobile devices in the sensing area increase. This relates to the increase in the number of mobile devices capable of covering one location. For improved global energy-efficiency, both algorithms select a larger set of active mobile devices to cover application requirements, which decreases the energy gain for Algorithm Info-Aggregation, despite the difference in the number of active mobile devices offloading sensed data between the two algorithms. Further data analysis and model implementation for improved energy-efficiency will be done as part of future work of this dissertation.
Fig. 3.5 shows the mean percentage difference in the cumulative volume of data available to be offloaded for Algorithm Info-Aggregation in comparison to Algorithm No-Aggregation for both static and dynamic mobility patterns. As well as leading to significant energy savings this reduction in the volume of offloaded data may also have a positive impact in terms of the efficient use of the limited bandwidth available to mobile devices.

3.6 Conclusion

The aim of the first research questions (RQ1) stated in 1.1.1 was to define the architecture and placement of a sensing framework. This framework would be responsible for interacting with multiple applications to gather their sensed data requirements, connecting with mobile devices present within the requested physical areas and creating a mapping of available sensor resources with the corresponding application requests. Once such a framework was defined, the second research question(RQ2) stated in 1.1.2 focused on reducing the redundancy that is possible due to overlapping streams of sensed data being offloaded by mobile devices to satisfy multiple application requirements. This is essential to improve energy-efficiency of such systems and avoid congestion due to large amounts of offloaded sensed data.

This chapter answers these questions by introducing the characterization of a collaborative sensing framework that supports sensor-driven mobile applications to transform the way in which users interact with their physical environment. Furthermore, it proposes a novel technique for sensor data collection for mobile cloud computing based on the framework that aggregates information and serves it to multiple applications. Algorithm Info-Aggregation is the scheduling algorithm deployed on the framework that identifies the potential of multi-tasking the capabilities of a mobile device and reduces the number of actively transmitting mobile devices.

Thus, the main contribution of this chapter is in the form of Algorithm Info-Aggregation that successively applies a revised version of frequent pattern mining to find a trade-off between energy efficiency, sampling rate of sensed data requested by applications and the volume of data offloaded from mobile devices. This helps in building the system and making it capable to harness the advantage of embedded sensors in mobile devices for bigger context-aware applications.

Chapter 4

Application-specific State Machines

4.1 Introduction

Embedded sensors in mobile devices have increasingly become more sophisticated, covering a rich set of sensing capabilities, that include environmental sensors (barometers, photometers, and thermometers), motion sensors (accelerometers, gravity sensors and gyroscopes) and positional sensors (compasses and magnetometers). This has lead to a rise in mobile applications that rely on these sensors for providing a personalized, context-aware experience to the users. With the ubiquity of sensor devices that have been predicted to proliferate under the Internet of Things paradigm, this grows multi-fold. However, continuous sensing and offloading of large volumes of mainly redundant sensor data for multiple applications has been known to deplete mobile device resources quickly. As such, intelligent systems are required to decide when sensing and offloading operations should be performed along with supporting quick localization of the contextual change. Furthermore, interactions between the static IoT sensor devices and embedded mobile device sensors must also be quantified. These challenging questions have been documented as the third (RQ3) and fourth research question (RQ4) in Chapter 1.

This chapter addresses these questions by presenting an application-specific state machine approach for the collaborative sensing framework that encodes all application requirements. These state-machines are uses to control the rate of offloading sensed data to the server and are maintained for small regions of the physical area of interest. This study has been completed in an iterative manner and is presented in the following sections.

In section 4.2, simplifying assumptions for focus target sensing are made and opportunistic sensing techniques are studied for enterprise applications where mobile handsets, provided by the organisation, are used to provide context information via the centralized framework to enterprise cloud hosted applications that monitor and control the work environment. This data offloaded by the devices is processed in the mobile cloud which removes not only the limitations of processing but also the need for installation of specialized sensors. The algorithm Context-Localize, presented in the subsections focuses on identifying and quickly reacting to contextual change by utilizing physical known information regarding the enterprise building and the application properties encoded in a simple state machine. The section concludes by evaluating the algorithm and presenting key experimental results from the analysis.

After learning these capabilities of the state machine, the next section presents a more general IoT-inspired setup with multiple applications and state machines. The mobile devices assume the role of gateways in this situation, since they now access the various static sensor devices and interact with multiple applications requesting for sensed data with different reporting rates. Removing the notion of continuously reporting the sensed data, the problem formulation presented in Chapter 3 is extended and intelligence is given to the framework to identify the event-driven nature of mobile sensing applications. An improved algorithm, Assisted-Aggregation is thus designed which makes use of consolidated state machines and shows an improvement in terms of residual energy of the mobile devices, number of devices actively offloading and the volume of the offloaded data.

This chapter concludes by presenting a summary of the research and its applicability with respect to the research questions. This work has been disseminated in Loomba et al. (2014a) and Loomba et al. (2017).

4.2 An Environment Monitoring Enterprise Application Scenario

Without considering the deployment of dedicated IoT sensors, this subsection harnesses the collaborative sensing framework to support mobile devices within an enterprise network to opportunistically identity and localize the changing context in the environment. Two applications are considered for this experiment namely odour detection and fire detection for which sensed data mainly includes sampling of the levels of a particular gas, the humidity in the environment, the temperature and other environmental factors. Additionally, a mobility model is not simulated and the framework periodically checks for disruption in sensing signals.

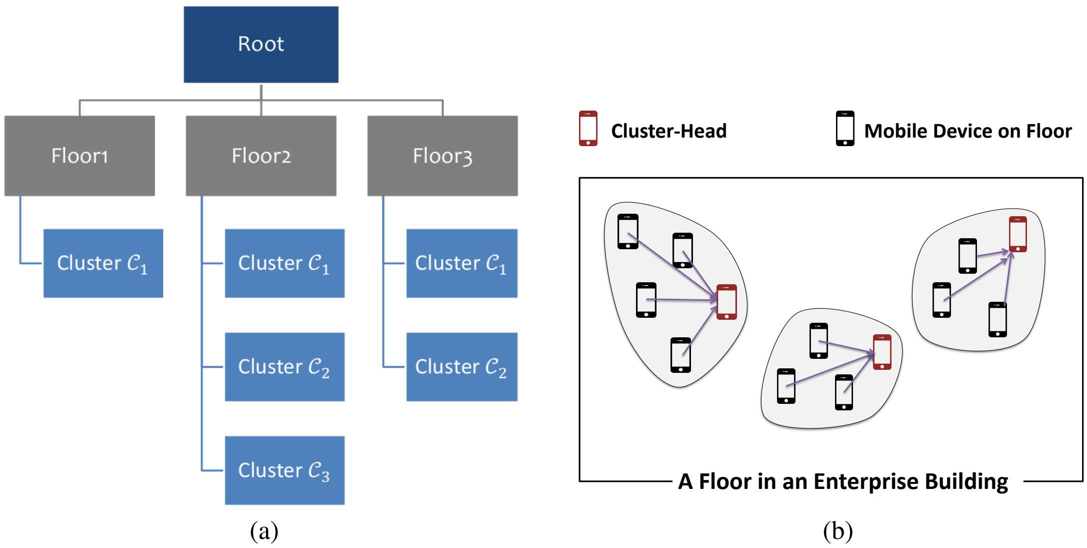


Fig. 4.1 Hierarchical Clustering Model presented in §4.2 for an enterprise building scenario: (a) shows the hierarchical cluster structure for a building with 3 floors, each floor cluster is divided using the k-Means clustering algorithm, depending on the number of rooms on the floor; (b) shows clustered mobile devices on a floor with the designated cluster-head.

This section is organized as follows. First a multi-tier cluster arrangement of the devices is presented, followed by the introduction of an application-specific state machine and the algorithm design. Next, the algorithm is evaluated based on a simulation model and finally, experimental results are presented.

4.2.1 Clustering Framework

This subsection presents the simple formulation of a clustering framework with $|\mathscr{X}|$ clusters to organize the mobile devices in a hierarchical structure. Physical known building statistics are availed for its creation, which is also depicted in Figure 4.1a. Each floor of the building forms a separate cluster which gets divided into smaller clusters as depicted in Figure 4.1b. A mobile devices $n \in \mathscr{N}$ is placed within a cluster $\mathscr{C}_i$ in time interval k , where i defines the index of the cluster, depending on its 2D coordinates. The coordinates are represented as $(x_n(k), y_n(k))$ and defined as a function of the time interval. Each time interval is of duration Δt seconds, over the planning horizon $\mathscr{T}$. For $t \in \mathscr{T}$ and $k \in [1, \mathscr{T}/\Delta t]$, the time intervals are indicated by $(k-1)\Delta t < t < k\Delta t$. It is imperative that the placement and orientation of the mobile device do not affect its sensing accuracy. As such, the device uses a self-characterization property to indicate itself (Kurz et al., 2011) and automatically gets connected to the enterprise network which is used to determine its coordinates. The

framework uses WiFi positioning or other technologies provided by access point vendors, which are accurate within metres, and suffice for the applications under consideration. Next, the mobile devices on a given floor are readily identified since then the height of each floor is known. For dividing the floor-level cluster, the k-Means clustering algorithm is run which automatically divides the mobile devices into φ clusters per floor based on their relative location. The value of φ is dependent on the number of rooms in each floor. The Euclidean distance metric is chosen for this clustering algorithm as shown in Equation 4.1, where the distance between two mobile device n_1 and n_2 is calculated for time interval k. Thus all clusters $\mathscr{C}_i, \forall i$ are determined.
Additionally, each cluster $\mathscr{C}_i$ is represented by a leader or cluster-head $C_i \in \mathscr{C}_i$ of the mobile devices present in that cluster. This cluster-head is randomly selected in each interval to ensure that a single mobile device does not incur all communications costs which is assumed to be the same across entire time duration. Improved selection methodologies for cluster-head selection based on location information have been presented in Chapter 5. The sensed information for the cluster is the highest sensed information received from its sub-clusters or sensed by the cluster members within an interval k.

$$d(n_1,n_2)_k = \sqrt{(x_{n_1}(k) - x_{n_2}(k))^2 + (y_{n_1}(k) - y_{n_2}(k))^2} \tag{4.1}$$

To incorporate the mobility of mobile devices, location information of the device received by location-based WiFi services or WiFi-positioning is automatically updated for time interval k. Each time interval k has its own specific rate to update the locations during the day, dependent on statically processed results from the movement patterns of previous days.

4.2.2 Simple Application Specific-State Machine

Instead of adopting a continuous sensing and reporting method, an application state-machine driven methodology is introduced which involves collecting and processing information at different rates during the time horizon. This rate is the frequency at which the mobile devices sense and offload sensor data and it is assumed that each state of the state machine has an own associated sensing rate, represented by ξ_i where i is the index of the state. Thus, using the state-machines, the sensing rate increments/decrements and sensed data is offloaded only on transition to a new state. This helps the algorithm consume less energy, both while sensing and reporting.
In the clustering framework, each sub-cluster informs the parent cluster of the current state of the state machine and the highest sensed value. The state machine maintained for each

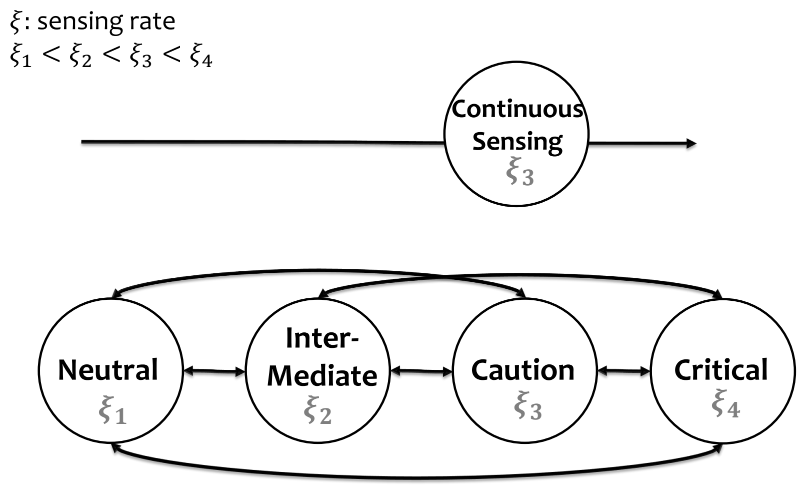


Fig. 4.2 Representation of a simple state-machine, with defined sensing rates for four states, namely neutral, intermediate, caution, and critical. This also supports a comparison of the sensing rate ξ_3 for continuous sensing, with the four state-driven rates, namely $\xi_1 < \xi_2 < \xi_3 < \xi_4$ in the state machine.

floor of the building is then set at the highest state collected from its sub-clusters. Thus when clustering is done for the next interval, the state machine for the floor is transferred to the sub-clusters without loss of information. By sensing, the sub-clusters can successively set the state machine to the state which represents its surrounding environment.

For simplification, an application state machine, as depicted in Fig. 4.2 with four states namely neutral, intermediate, caution and critical, is considered and a higher state is associated with a higher sensing rate. These states depict the effect of the percentage of carbon dioxide, carbon monoxide, methane or any other gas in the air as set by the monitoring application. Thus, the system moves from the neutral to the critical state depending on the sensed information.

4.2.3 Algorithm Design

This subsection defines the algorithm Context-Localize (Algorithm 4.1) which explains how the contextual change is identified whilst saving energy expended in the process.

For every time interval k, depending on the rate of location updates, the position of each mobile device $n \in \mathscr{N}$ is determined. It should be noted that the devices are assumed to be static between any two position updates. Using these positional coordinates, the mobile

Function 4.1 Algorithm Context-Localize

Require: Time Interval k

1: $\mathscr{X}$ = GETCLUSTERS(k);
2: SETCLUSTERHEAD($\mathscr{X}$);
3: **for all** $\mathscr{C}_i, \forall i$ **do**
4:     C_i=$\mathscr{C}_i$.getLeader();
5:     sensedValue=senseInformation(C_i);
6:     stateOfCluster= $\mathscr{C}_i$.setState(sensedValue);
7:     **if** stateOfCluster!=critical **then**
8:         updateSensingRate(stateofCluster)
9:     **else**
10:         localizeChangeInContext();
11:         informAuthorities();
12:         updateSensingRate(critical);
13:     **end if**
14: **end for**
15: **procedure** GETCLUSTERS(k)
16:     readPositionOfMobiles();
17:     $Floors[]$=createClusterForEachFloor();
18:     **for all** $Floor$ in $Floors[]$ **do**
19:         Statically assign mobile devices to the floor
20: according to its location
21:         K-MEANS($Floor$,φ)
22:     **end for**
23: **end procedure**

devices are organized in the hierarchical structure as defined in the procedure *getClusters()* from lines 15 to 22. In line 15, the mobile device is statically assigned to a floor-cluster according to the known floor-height. Then, k-Means is used to create φ sub-clusters within the floor-cluster in line 21. The parameter φ is equal to the number of rooms on the floor. Once clustering is completed and all clusters $\mathscr{C}_i, \forall i$ have been made, the cluster-heads are randomly selected. Using the function *senseInformation()*, either each mobile device in the cluster reports its sensor readings to the cluster-head or sensed data is collected from the cluster-heads of the sub-clusters. The state of the cluster is determined according to the highest sensor reading available with the cluster-head using the function *setState()* in line 6.

The algorithm responds to each state by either updating the sensing rate of the cluster as shown in line 8 or in case of a critical reading, by localizing the contextual change and delivering intimation messages to the associated authorities as stored in the framework, as shown in line 10 to 12. This localization of the environmental condition is possible by

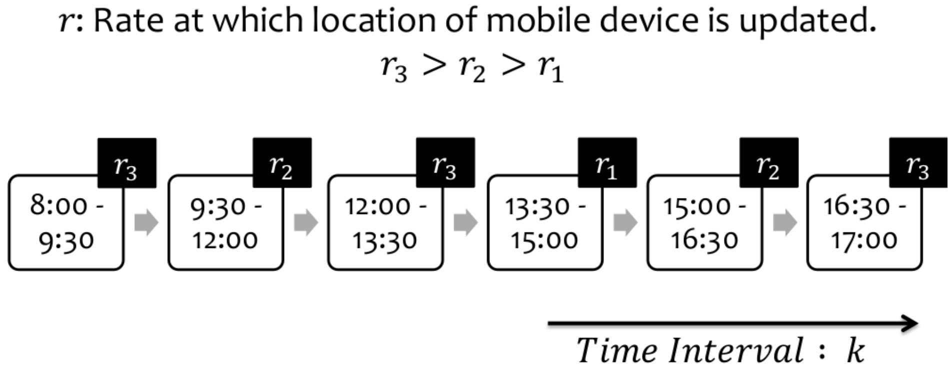


Fig. 4.3 Time interval dependent simulated rates for updating the location of mobile devices within the enterprise building.

examining the sub-clusters in a BFS fashion until the mobile device reporting a critical state is identified. Consequently, the system continues sensing and as the environment changes, it moves back to the caution state, then to the intermediate state and finally neutral state.

4.2.4 Evaluation

This section evaluates the performance of Algorithm Context-Localize (Algorithm 4.1) to calculate the percentage of energy savings for all mobile devices in comparison to a continuous sensing approach. To achieve this, the total number of mobile devices, the number of floors and rooms in a building, and the metric square meter of the enterprise building are varied. The section is begins by detailing the simulation model and all the parameters used to define the state machine, presents experimental results, and concludes by analysing the findings.

In the simulation, the flow of information is recorded which happens when a mobile device detects a change in its context by using Algorithm Context-Localize (Algorithm 4.1), the proposed algorithm for fire-detection or odour-detection. Currently, the algorithm outputs the chain of events as they happen and detects further information regarding the cause of change in context for the caution and critical state of the system. Table 4.1 documents the information available in the system for the various states of the system. The mobile device, its user identification code and the location is made available so that further action can be taken.

4.2.4.1 Simulation Model

The enterprise building is defined using the length and breadth of its base, for which two cases are considered namely a square base ($50m \times 50m$) and a rectangular base ($150m \times 50m$). Time progression within the simulation study is modelled in discrete time steps of duration $\Delta t = 30$ minutes.

Since environmental monitoring applications such as fire detection and odour detection are considered, the states of the application-specific state machine are calibrated accordingly. It is also ensured that all sensor readings are normalized to the metric system. The state machine for the fire detection application is simulated in the experiments based on the Discovery CO/Heat detector by Apollo (Apollo) that uses sensor readings regarding temperature and percentage of carbon monoxide in the air to detect fire. For the odour detection application, the percentage of carbon dioxide in the environment analogous to 1%, 3%, 5% and 8% are used. Sensing rates associated with the state machine depict the frequency with which mobile device updates the sensed reading. This rate increases as the cluster goes into a higher state and lowers when it goes to a lower state. The sensing rate is simulated to be equal to the sensing rate at the caution state. This allows more frequent readings in the caution and critical state of the system. To simulate a critical reading, a random location in the building was given an increasing function for sensed values of environmental readings which can be associated with the percentage of carbon monoxide and temperature in the environment in the case of fire detection or percentage of carbon dioxide in the case of odour detection.

A mobile device $n \in \mathcal{N}$ in an enterprise building is uniquely identified using an integer id or user number and random generators are used to identify its positional coordinates. This user number is associated with the person carrying the mobile and helps when intimation messages are sent to identify the mobile user. The number of sensors available within a mobile device are also randomly distributed.

Figure 4.3 depicts the varying time periods during the day along with the related rate of updating mobile device locations that are defined for this simulations. This is based on observations made for the research building where these experiments were simulated. It was noted that the position of the mobile users changed drastically in the lunch hour. This meant that during that time, location updates are made at a faster rate compared to the morning time slots. Also, meetings are scheduled in the evening so the rate is different as compared to morning slots as location of multiple mobile devices changes simultaneously for a fixed interval of time. This understanding helps in getting updated location clusters of the mobile devices during the day. Currently, the framework cannot learn this over time and uses static processing to set the various time periods.

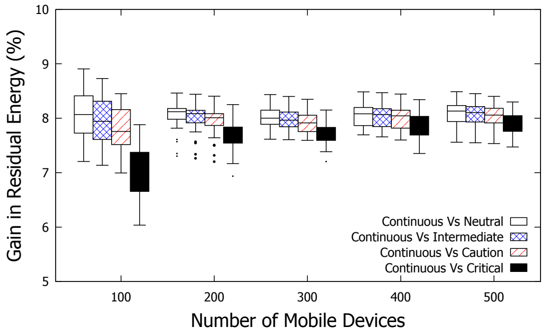
4.2.4.2 Results and Analysis

Each Java-based simulation is specified by the number of mobile devices, number of floors and area of the enterprise building. The total number of mobile devices $|\mathscr{N}|$ sensing the environment is varied over a range from 100 to 500 and the number of floors is assumed to be in $\{5, 10, 15\}$. Each experiment uses random generators to distribute the sensor-types in the mobile device and thirty runs of the experiments are done.

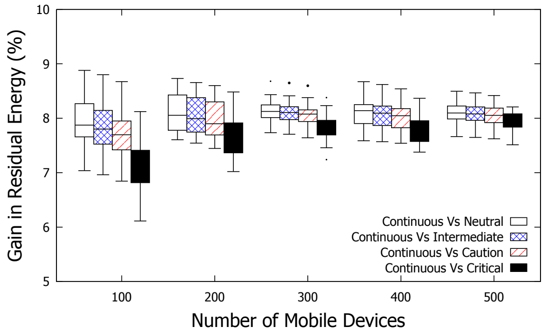
The energy performance of the Algorithm Context-Localize (Algorithm 4.1) when adopting state-driven sensing and reporting rate is compared to a continuous sensing and reporting approach. For comparative analysis, percentage difference between these approaches are recorded for different states of the environment. The computed box-plots for the two cases are presented in Figure 4.4. It can be seen that irrespective of the reported state, Algorithm 4.1 is more energy efficient and more than 5% of the residual energy can be saved. Since, the neutral state of the environment will occur more frequently, even more energy saving can be reported by modulating the sensing rates. Also, when comparing the critical state of the environment with the continuous approach, energy saving are observed, despite the higher sensing rate. This is attributed to the fact that the system launches into the critical state slowly and energy is saved in the neutral and intermediate states for the initial time intervals.

Figure 4.5 shows the energy savings made when different number of floors are considered for the enterprise building and the total number of mobile devices are specified as 300 and 500 device. It can be discerned that the number of floors in an enterprise building do not affect the energy saved to a significant extent. Additionally, Table 4.1 documents the information available in the system for the various states of the state machine.

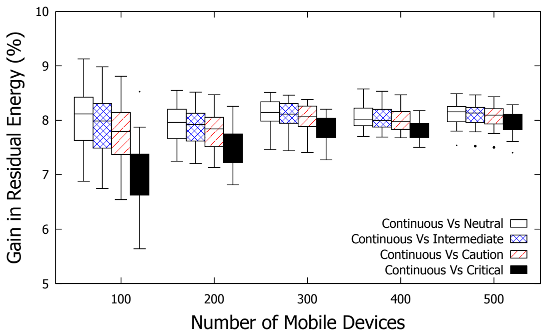
Thus, this study allows a quick analysis of the state machine approach and is scaled in §4.3 for an IoT scenario with multiple application requirements.



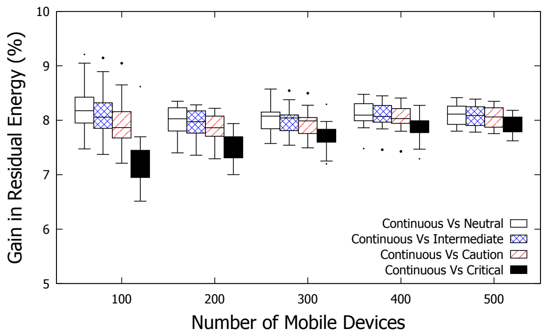
(a) 5 floors, Square Base



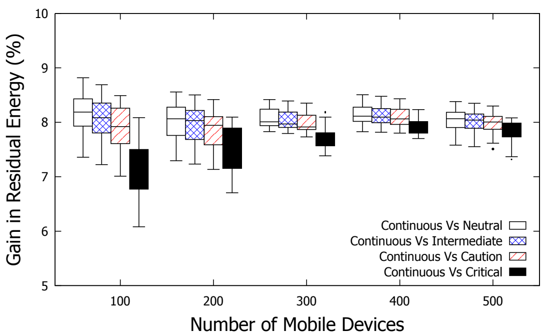
(b) 5 floors, Rectangular Base



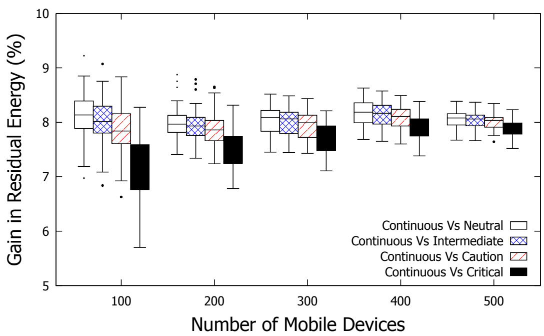
(c) 10 floors, Square Base



(d) 10 floors, Rectangular Base

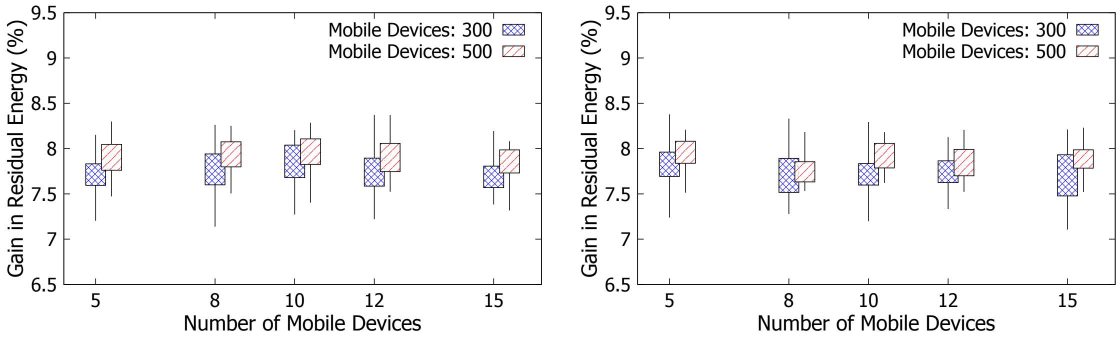


(e) 15 floors, Square Base



(f) 15 floors, Rectangular Base

Fig. 4.4 Comparative percentage difference in the cumulative residual energy held in mobile device batteries between a continuous sensing approach and Algorithm Context-Localize vs. the total number of mobile devices in the sensing environment over three cases using 5, 10 and 15 floors respectively for a square base area of 50×50 (a,c,e) and a rectangle base area of 150×50 (b,d,f). Each graph shows the percentage gain in residual energy when different states (neutral, intermediate, caution and critical) are reported by Algorithm Context-Localize.



(a) 50×50m base (b) 150×50m base

Fig. 4.5 Comparative percentage difference in the cumulative residual energy held in mobile device batteries between a continuous sensing approach and a critical state recorded by Algorithm Context-Localize across different number of floors in the enterprise building. The number of mobile devices are taken to be 300 and 500, for (a) a square base of 50×50m and (b) a rectangular base of 150×50m.

Table 4.1 Simulation output for different environmental states recorded by Algorithm Context-Localize.

State Of Environment	Output from Simulation
NeutralState	Root Cluster of Building reported a ‘NEUTRAL’ state.
IntermediateState	Contextual change observed. Root Cluster of Building reported ‘INTERMEDIATE’ state.
CautionState	Contextual change observed. Root Cluster of Building reported ‘CAUTION’ state. Enterprise Building is in ‘CAUTION’ mode.
CriticalState	Contextual change observed. Root Cluster of Building reported ‘CRITICAL’ state Detecting the location of potential problem. Identifying the reporting mobile device/cluster-head. Cause of problem is identified in Cluster 9 Cluster-Head is Mobile ID: 27, with USER ID: 20057666 Informing all mobile devices in Cluster 9. Intimation Messages sent to authorities.

4.3 Multiple Internet-of-Things Applications scenario

Minimizing communication costs and the energy expended by mobile devices during sensing and reporting continues to be an important technical challenge. Extending the application-specific state machine approach presented in §4.2 of this chapter, this section presents an evolved collaborative framework for an IoT-inspired scenario, where the mobile devices act like gateways, with access to multiple static sensor devices that are encountered in their path. This implies that a dynamic set of available sensors is linked with the mobile devices. Furthermore, the framework presented in Chapter 3 is given the intelligence to identify the event-driven nature of mobile sensing applications and the challenges of satisfying multiple applications requests for sensor data relating to different physical area(s), minimum accuracy requirements and reporting constraints are also studied.
This section first presents a complete problem formulation, expanding on §3.3 by defining IoT sensors and the concept of creating consolidated state machines for previously-identified areas depicting thresholds specified by multiple applications. Next the design of Algorithm Assisted-Aggregation 4.2 is discussed that applies frequent pattern mining to identify the best combination of embedded sensors in mobile devices and available IoT static sensors to simultaneously satisfy the requests of multiple applications, whilst reducing volume of offloaded data and the number of devices that actively offload information into the cloud to save energy. Additionally, depending on the size of the consolidated state machine, the algorithm factors it into independent smaller state machines, supporting scalability issues. The section concludes by providing experimental results of this research study.

4.3.1 Problem Formulation

This section presents a mathematical formulation for the deployment of the application-specific state machine approach in a smart-city scenario, similar to Latré et al. (2016) and Rathore et al. (2016). Fig. 4.6 shows the settings with connected infrastructure components and services like education, healthcare, transportation etc. along with secure smart housing that interact with the application cloud server via the centralized framework and Table 4.2 lists all notations used for the formulation. By examining the city as a collection of interest areas, the framework is able to monitor and respond to the ever-changing dynamic of the city. Several context-aware applications actively offload and process sensor data relating to either environmental factors like temperature, humidity, pressure and unwanted gas levels, or concerning the rising noise issues for traffic management or promoting social interactions (Khan et al., 2013) for this purpose.

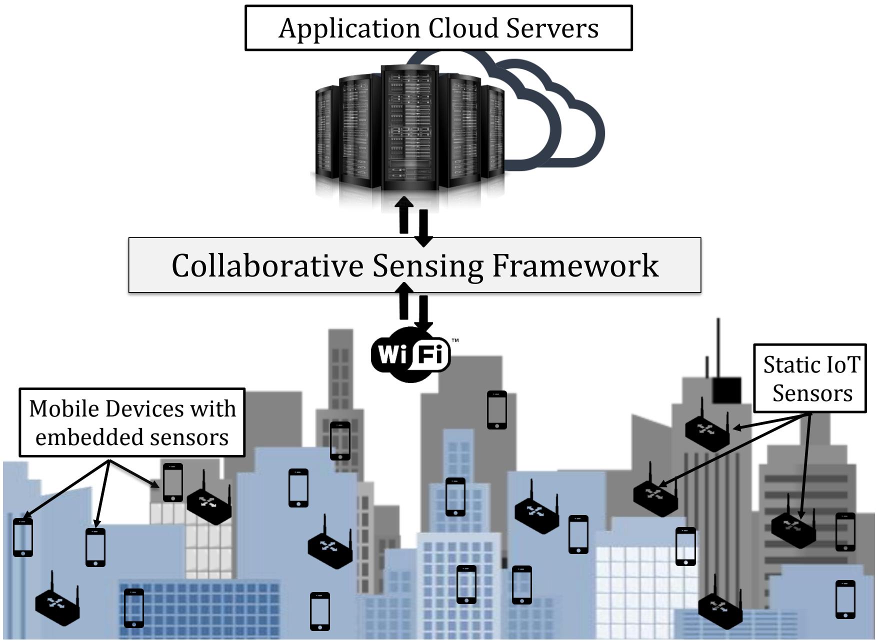
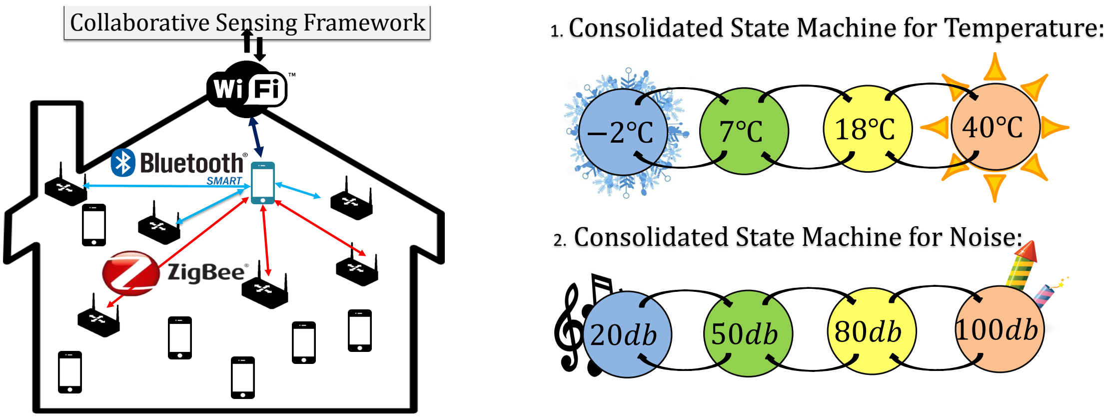


Fig. 4.6 Extended Architecture Representation of the Collaborative Sensing Framework for a smart-city scenario, facilitating seamless interaction between the application cloud, static IoT sensors and mobile devices using WiFi across multiple connected infrastructure components and services like education, healthcare, transportation and smart housing.



(a) (b)

Fig. 4.7 Extended Mobile Sensing Model for the collaborative sensing framework that includes static IoT sensors and state machine representations: (a) shows how one request location is covered by an active mobile device using its own embedded sensors and available static IoT sensors in the request location whilst offloading the collected sensor-data to the cloud for processing using WiFi, when required; (b) shows two consolidated state machines for sensor-types temperature and noise present for a request location.

With added understanding of the event-driven nature of application in contrast to Chapter 3, this centralized framework is now capable of simultaneously connecting to the numerous static sensors present within the interest areas and varied embedded sensors of the mobile device to satisfy application constraints. The ability of a mobile device to act as a travelling agent is also leveraged and an aggregation plus transmission schedule is calculated by the framework to instruct devices to accurately collect sensor data either from its own sensors or from the nearby static sensors using BlueTooth or ZigBee technologies. Depending on the reporting constraints for the applications, anonymized sensor data streams are offloaded to the application cloud for further processing and storage. Such an architecture supports both participatory and opportunistic sensing (Lane et al., 2010) as explained in Chapter 3.

4.3.1.1 Terminology

The problem is formulated for a physical sensing area $\mathscr{L}$, composed of a set of square interest areas or request locations for which sensed data is requested by the applications over a duration of $\mathscr{T}$ seconds. An individual request location $\mathscr{L}_i \in \mathscr{L}$ is a set of all 2D coordinates that define the ith location in the sensing area. After every Δt seconds, pre-processing of the collected sensor data is done to ensure that all requirements of the applications have been met. For $t \in \mathscr{T}$ and $k \in [1, \mathscr{T}/\Delta t]$, these time intervals are denoted by $(k-1)\Delta t < t < k\Delta t$. Two kinds of sensors have been modelled in this study that cover a range of different sensor-types denoted by set $\mathscr{H}$, namely embedded sensors within the mobile device and static sensors that have been installed in the request locations. It is assumed that the sensed information present for any sensor does not vary much during a time interval and also does not affect how application requirements are addressed.

For each mobile device n in $\mathscr{N}$, $\mathscr{M}_n \subseteq \mathscr{H}$ defines the set of unique sensor-types embedded within the device. The individual sensor-type $m \in \mathscr{M}_n$ can sense data with accuracy ρ_m^n and accumulate v_m^n volume of data (if activated) in every time interval. The request location covered by the mobile device is determined by the 2D position coordinates of the device $(x_n(k), y_n(k))$, defined as a function of the time interval k. Additionally, $\mathscr{N}_i^k$ represents the set of all devices present in a request location, mathematically represented by $\forall n \in \mathscr{N}_i^k, s.t.(x_n(k), y_n(k)) \in \mathscr{L}_i \in \mathscr{L}$ for time interval k.

Each static sensor, $s \in \mathscr{S}_i$ for request location $\mathscr{L}_i \in \mathscr{L}$ contains $\mathscr{M}_s \subseteq \mathscr{H}$ unique sensor-types. Each individual static sensor-type $m \in \mathscr{M}_s$ can sense with accuracy ρ_m^s and accumulate v_m^s volume of data (if activated) for every time interval. These static sensors do not directly connect to the framework and only interact with the mobile devices present in the request location for security reasons. This implies that only those request locations

Table 4.2 Notation used for Problem Formulation of Chapter 4.

Notation	Description
$\mathscr{L}$	Set of Request Locations in the sensing area
$\mathscr{L}_i$	Set of 2D coordinates for the ith Location
$\mathscr{T}$	Planning horizon
t	Time variable
Δt	Time Interval for aggregation
k	Index of a time interval
$\mathscr{H}$	Set of universal sensor-types
h	Index of universal sensor-types
$\mathscr{N}$	Set of mobile devices
n	Index of mobile devices
$\mathscr{M}_n$	Set of sensor-types on mobile device n
m	Index of sensor-types in a set
ρ_m^n	Scalar accuracy value of sensor-type m in the mobile device n
v_m^n	Volume accumulated by sensor-type m in the mobile device n
V_n^k	Total Volume accumulated by mobile device n in time interval k
$(x_n(k), y_n(k))$	Coordinates of the mobile device in time interval k
$\mathscr{N}_i^k$	Set of mobile devices in request location $\mathscr{L}_i$ for time interval k
$\mathscr{S}_i$	Set of static sensors for request location $\mathscr{L}_i$
s	Index of static sensors
$\mathscr{M}_s$	Set of sensor-types on static sensor s
ρ_m^s	Scalar accuracy value of sensor-type m in the static sensor s
v_m^s	Volume accumulated by sensor-type m in the static sensor s
$\mathscr{A}$	Set of applications
a	Index of applications
$\mathscr{M}_a$	Set of sensor-types requested by application a
M	Index of sensor-types requested by application a
$\mathscr{L}_a$	Set of request locations of application a constant in time
${\rho^*}_M^a$	Minimum scalar accuracy of sensor-type M for application a
τ_M^a	Set of sensing time points of sensor-type M for application a
$\mathscr{Z}_h^i$	Set of states for sensor-type h for request location $\mathscr{L}_i$
p_{mn}^k	0-1 variable to indicate if a sensor-type m is sensed from mobile device n during time interval k

Notation	Description
q_{ms}^{k}	0-1 variable to indicate if a sensor-type m is sensed from static sensor s during time interval k
y_n^k	0-1 variable to indicate if a node is activated during time interval k
E_n^k	Energy consumption for collecting sensor data for device n in time interval k
F_n^k	Energy consumption for storing and accessing a factored state machine for device n in time interval k
G_n^k	Energy consumption required for locally pre-processing data, communication and offload of the sensor data for device n in time interval k
ϕ_1, ϕ_2	Constant Values
ζ	Minimum percentage of battery level
B_n	Full batter level of node n
b_n^t	Battery level of node n at the start of time interval k
γ, δ	Normalization parameters to balance battery consumption, active mobile devices and volume offloaded into cloud servers

which have at least one mobile device present for the entire duration of the kth time interval can be sensed for any application. Each request location is taken to meet this constraint and more complex scenarios will be studied as future work. Fig. 4.7a depicts how one mobile device covers the request location (for e.g. a house) by using its own embedded sensors and the surrounding static sensors via BlueTooth or ZigBee communication technologies whilst offloading the collected sensor-data to the cloud for processing using WiFi, when needed. The sensor-types requested by application $a \in \mathscr{A}$ are defined by the set $\mathscr{M}_a \subseteq \mathscr{H}$ for which sensing is required for every request location in the set $\mathscr{L}_a \subset \mathscr{L}$. Applications also specify minimum accuracy requirements $\rho{*}_M^a$ and a constant sensing time-interval which can be translated into a set of time-points τ_M^a for each sensor-type $M \in \mathscr{M}_a$. The time interval k then contains all the sensing time-points for all applications: thus $\bigcup_{a\in\mathscr{A},M\in\mathscr{M}_a}(k-1)\Delta t < t \in \tau_M^a < k\Delta t$ indicates the applications and the sensor-types for which sensed data is requested in the time interval k.

However, it is examined that applications do not require updated sensor-information at every time-point in τ_M^a as is contingent to the understanding that most applications make decisions depending on the information derived regarding the context of a particular location, which is composed of individual raw sensor readings. As presented in §4.2 continuous offloading does not serve much purpose and application specific machines with context encoded in each state along with a reporting threshold required for that sensor-type are used. This is beneficial as it supports seamless identification of the information required by the application. This instructs the framework to collect and pre-process the data according to time-points in set τ_M^a and to transmit updated sensor data to the cloud only when it crosses a reporting threshold similar to the state machine shown in Fig.4.2. For example, when an application requests for temperature readings to be sensed at a time-interval of 10 minutes, but reported only when the temperature value crosses 7 °C or 18 °C. This information is used by the framework to create consolidated state machines from all applications requesting for sensor information from each location. The state machine for sensor-type $h \in \mathscr{H}$ from location $\mathscr{L}_i$ is denoted by a set of states $\mathscr{Z}_h^i$ where each state is composed of the threshold value for that state, the state-transition rules and the transmission rules for offloading data into the cloud. In contrast to the state machine presented in 4.2.2, only neighbouring state transitions are considered for the new consolidated state machines and dynamic changing of sensing rates is not supported. Fig. 4.7b depicts two consolidated state-machines for sensor-type temperature and noise maintained for a request location by the framework. At the end of each time interval k, the reporting mobile device pre-processes the data and checks with the state machine to decide whether it should offload the data or not.

4.3.1.2 Problem Statement

Given the assumptions outlined above, the optimization problem focuses on a) optimizing the selection of mobile devices to offload sensed information into the cloud, b) reducing the global energy consumption of the mobile devices and c) reducing the volume of data being offloaded into the cloud.

4.3.1.2.1 Decision Variables Since multiple sensors are available for each sensor-type, two decision variables are used to identify which sensor is being used for accessing the data for one sensor-type m within a time-interval k. These are denoted by $p_{mn}^k \in \{0,1\}$ for every mobile device n and $q_{ms}^k \in \{0,1\}$ for every static sensor-type s. Additionally, for fairness and scalability, the system ensures that a single mobile device is not always reporting and storing the entire consolidated state machine for each request location. Clustering approaches that help select energy-efficient cluster-heads as presented in Chapter 5 will be explored for this purpose in future work.

The framework next needs to decide a set of candidate reporting devices and factor the state machine, so that each candidate reporting device has a part of the state machine. For this purpose, a mobile device is considered active if it is responsible for reporting the sensed data to the application for one or more sensor-types during time interval k. This is indicated by the decision variable $y_n^k \in \{0, 1\}$ which is equal to one if the device is a candidate reporting device.

4.3.1.2.2 Constraints This subsection defines the constraints relating to the sensor coverage area, accuracy and minimum device battery level as follows. These constraints are similar to the constraints defined in §3.3 but the formulation is updated for inclusion of the accessible static IoT sensors.

Coverage Constraint: Each application a has specified sensor-types $\mathscr{M}_a$ for request locations $\mathscr{L}_a$ that need be covered for time interval k. This constraint is defined in Eq. 4.2 to ensure that the request locations have the specified sensors and at least one mobile device that can offload sensed data when required.

$$\begin{aligned}
&\forall k \in [1, \mathscr{T}/\Delta t], \forall a \in \mathscr{A}, \forall \mathscr{L}_i \in \mathscr{L}_a, \forall M \in \mathscr{M}_a \cap (\mathscr{M}_s \cup \mathscr{M}_n), \\
&\forall n \in \mathscr{N}_i^k, \forall s \in \mathscr{S}_i : \\
&\mathscr{N}_i^k \neq \emptyset \\
&\textstyle\sum_{\mathscr{N}_i^k} y_n^k \geq 1
\end{aligned} \tag{4.2}$$

Accuracy Constraint: The accuracy constraint for each application must be met by either the embedded sensors on the mobile device or the available static sensors that can be accessed by the mobile device for each request location. This is represented in Eq. 4.3

$$\begin{aligned}
&\forall k \in [1, \mathscr{T}/\Delta t], \forall a \in \mathscr{A}, \forall \mathscr{L}_i \in \mathscr{L}_a, \\
&\forall n \in \mathscr{N}_i^k : y_n^k = 1, \forall s \in \mathscr{S}_i, \forall M \in \mathscr{M}_a \cap (\mathscr{M}_s \cup \mathscr{M}_n) : \\
&\rho{*}_M^a \leq max(\{\rho_M^n\} \cup \{\rho_M^s\}) \\
&p_{mn}^k = 1 \implies \exists k : \rho{*}_M^a = \rho_M^n \\
&q_{ms}^k = 1 \implies \exists k : \rho{*}_M^a = \rho_M^s
\end{aligned} \tag{4.3}$$

Battery Life Constraint: Energy costs are incurred by a mobile device n related to its monitoring and sensing activities. E_n^k denotes the energy consumed to collect sensor data from all the embedded-sensors and from the surrounding static sensors during time interval k.

Additionally, the device loses further energy if it is selected as a candidate reporting device for request location $\mathscr{L}_i$ and needs to pre-process sensed data whose volume $V_n^k = \sum_{m\in\mathscr{M}_n} v_m^n \cdot p_{mn}^k + \sum_{s\in\mathscr{S}_i} \sum_{m\in\mathscr{M}_s} v_m^s \cdot q_{ms}^k$. First, it consumes energy to store and access a factored state machine or part of the consolidated state machine for each of the sensor-types, which is represented by $F_n^k = \phi_1 \cdot \left(\sum_{m\in\mathscr{M}_n} p_{mn}^k + \sum_{s\in\mathscr{S}_i} \sum_{m\in\mathscr{M}_s} q_{ms}^k \cdot y_n^k\right) \cdot \frac{|\mathscr{Z}_m^i|}{\sum_{\forall n\in\mathscr{N}_i^k} y_n^k}$. Next, it consumes energy to locally pre-process the data, identify the state which might involve communication with surrounding devices and offload the collected sensor data which is represented by $G_n^k = \phi_2 \cdot V_n^k$. The battery constraint below thus states that the battery level of the mobile device n at the start of time interval k, denoted b_n^k, is below ζ of the full battery level B_n. This restriction is imposed on all mobile nodes to ensure battery availability for monitoring, sensing, local data pre-processing and information offload into the cloud and defined in Eq. 4.4.

$$
\begin{aligned}
&\forall n \in \mathscr{N}, \forall k \in [0, \mathscr{T}/\Delta t]: \\
&b_n^k >= \zeta B_n \\
&b_n^k - E_n^k - F_n^k - G_n^k = b_n^{k+1}
\end{aligned}
\tag{4.4}
$$

4.3.1.2.3 Objective Function During each time interval k, a deployment has a set of possible mobile devices with embedded sensors and static-sensors installed within a location that are activated to provide information at different accuracies for applications. The objective is thus threefold:

- Minimize energy consumption due to sensing, local data processing and information offload into the mobile cloud. This is reflected in the first term which is a summation of the energy losses incurred in the process;
- Minimize number of active mobile devices in the sensing environment. This is represented by the second term which is the summation of the active mobile devices in the system to which a weight of γ is allotted;
- Minimize the volume of data offloaded into the cloud. This is represented in the third term which is allotted a weight of δ.

These can be expressed formally in Eq. 4.5.

$$
\text{minimize} \quad \sum_{k=1}^{\mathscr{T}/\Delta t} \left(\sum_{n\in\mathscr{N}} (E_n^k + F_n^k + G_n^k) + \gamma \sum_{n\in\mathscr{N}} y_n^k + \delta \sum_{n\in\mathscr{N}: y_n^k=1} V_n^k \right)
\tag{4.5}
$$

Algorithm 4.2 Algorithm Assisted-Aggregation

```
for k=1 to 𝒯/Δt do
    t.start=(k − 1)Δt
    t.end=kΔt
    for n =1 to 𝒩 do
        predictLocOfMobileDevice(n, t.start, t.end)
    end for
    AppSensorPairs[]=getAllAppSensorPairs(k)
    for AppSensor ∈AppSensorPairs[] do
        loc l[]=getLocs(AppSensor,k)
        NodeSets=CALCNODESETFORLOC(l[],k)
    end for
    Tree T=FP-TREE(NodeSets)
    FP P=FP-GROWTH(T.root,null)
    maxSizes[]= calcTwoHighestForMostFrequent(P)
    for Set pinP do
        if p.size ∈ maxSizes[] & hasBattery(p)=true then
            fSet=fSet ∪ p
        end if
    end for
    CALCAPPSENSOR(fSet, AppSensorPairs[])
    Run CALCENERGYCONSUMPTION(k)
    Run CALCPERCENTAGETOOFFLOAD(k)
end for
function CALCAPPSENSOR(fSet, AppSensorPairs[])
    for AppSensor ∈ AppSensorPairs[] do
        NodeSet=getSetWithMinDist(AppSensor, fSet);
        for all n ∈ NodeSet do
            reportThreshold[]= setReportingDevice(n)
            list = getNeighbours(n)
            rawValue= preProcess(n)
            if rawValue ∈ reportThreshold[] then
                offloadAccToState()
            else
                getStateFromNeighbours(list)
                offloadAccToState()
            end if
        end for
    end for
end function
```

4.3.2 Algorithm Design

This section describes the Algorithm Assisted-Aggregation 4.2 that focuses on reporting sensed data to multiple applications by multi-tasking the capabilities of each mobile device, whilst ensuring that the consolidated state machines of the various sensor-types, maintained for each request location, are factored and distributed between a set of reporting devices. This leads to a reduction in redundancy of offloaded data streams and supports dynamic altering of the reporting rate of mobile devices for better energy management.

Most terminology used in this section has already been presented in §3.4. This includes the term *Application-Sensor pair* represented by $< a, M >$ which uniquely defines the relation between an application and a sensor-type requested by the application. This helps in identifying the request locations $\mathscr{L}_a$, the minimum accuracy needed for the sensor-type $\rho{*}_M^a$, at sensing time-points τ_M^a with reporting thresholds based on an application-specific state machine. Additionally, a *NodeSet* refers to a set of mobile devices that can collectively be used to cover the sensing and reporting requirements for an Application-Sensor pair. For this set, energy consumption is defined as a summation of the individual energy consumed by the mobile devices and it is ensured that each mobile device satisfies the accuracy constraint.

This algorithm works in the following manner. First, the framework uses technologies such as GPS or WiFi Positioning to retrieve the starting position of each mobile device. A look-ahead approach is adopted for every time interval k to predict the mobility pattern of the mobile devices. The function *predictLocOfMobileDevice(n,t.start, t.end)* incorporates an instance of the mobility model and determines the location covered by the mobile device n during the specified time interval. Next, the set τ_M^a indicates the Application-Sensor pairs that request for sensor activities in k. When calculating all NodeSets that fulfil the requirements for an Application Sensor pair in line 10, a NodeSet is only considered if each mobile device in the set covers its current location for entire time interval k. It is assumed that the variance in the movement of the mobile device within the interval does not affect the sensing readings if it is still in a position to cover the location. By exploiting the possible communication of the mobile device with all static IoT sensors in proximity, the best sensor (embedded or static) meeting the accuracy constraint $\rho{*}_M^a$ is selected. Furthermore, a trade-off is made between the communication cost with the static sensors and the accuracy of the sensor data provided to the Application-Sensor pair. This means that the static sensors are only accessed if the embedded sensors in the mobile device are unable to satisfy the accuracy constraints.

The frequency of every subset of mobile devices amongst all NodeSets is calculated and to ensure that subsets containing devices which can sense and report for more than one application are selected. This is possible by deploying the FP-Growth algorithm (Han et al., 2007a) as Functions 3.3 and Function 3.4 which have been described in Chapter 3. As presented in detail in Chapter 3, the FP-Growth is based on the principle that when a set of $(w+1)$ mobile devices is frequent, then the subsets of w mobile devices will also be frequent. The proportion of times a set occurs in the interval is termed as *support* and a *minimum support* is required by a set to be termed as frequent. Using these definitions, the NodeSets are ordered in decreasing order of support count of the mobile devices and recursively added into a FP-Tree or frequent pattern tree in function 3.3. The recursion ensures that NodeSets sharing a common prefix are attached to a single child of the root node of the tree. Following this, function 3.4 is used to mine and extract frequently occurring subsets in a breadth-first fashion. The subsets where a specific mobile device is a suffix node are examined first and conditional trees are made with the prefix paths of this device. This transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. The output from these functions is present in the variable P in line 13.

This pattern mining technique (Han et al., 2004, 2006) thus identifies the largest and second largest frequently occurring mobile subsets. Developing from the concept described in §3.4, the *base subset* or $fSet$ is a concatenation of all the subsets that are equivalent in size to these sets and have adequate energy for sensing and reporting activities. Iteratively, one NodeSet is selected for each Application-Sensor pair that utilizes the maximum number of mobile devices in this base subset as defined in Function *calcAppSensor* in line 24. Every device that is a part of this selected NodeSet, is designated to be a candidate reporting device for the embedded sensor-type or for a static sensor whose data can be collected by the device. In line 28 and 29, the candidate reporting device receives a factored state machine with a range of reporting threshold stored in array $reportThreshold[]$ along with a list of the neighbouring reporting candidates stored in array $list[]$. This section only explores a simple parallel factoring technique to split the state machines, but more complicated techniques like those defined by Devadas and Newton (1989) will be explored as part of future work. After minimal pre-processing of sensor data for its location in line 30, the device checks whether the raw value lies within the reporting thresholds of its factored state machine. In case the value lies outside the range of the reporting thresholds, the device starts a one-2-many connection with its neighbouring reporting devices in line 34 to determine whether the data needs to be offloaded. The sensor data is offloaded using the function *offloadAccToState* accordingly. Finally, the energy consumed and the volume offloaded by each mobile device for the time interval k is calculated using function *calcEnergyConsumption(k)* and the local search heuristic defined in function *calcPercentageToOffload(k)* (presented in §3.4) respectively.

4.3.2.1 Complexity Analysis

Th worst case running time of the Algorithm Assisted-Aggregation is equal to the worst-case time complexity of Algorithm Info-Aggregation, presented in §3.4.3.
Assuming that the maximum number of request locations covered by all application is given $\forall a, |\mathscr{L}_a| = l$ with a maximum of $d <= |\mathscr{N}|$ mobile devices for the time interval k, then the number of NodeSets for each Application-Sensor pair will be equivalent to d^l. Next, the complexity of the FP-Growth Algorithm which is proportional to the number of unique elements $d \cdot l$ present in the header table and the depth of the FP-Tree. Thus, in the worst case, the tree is an unbalanced tree and its depth is upper-bounded by $d \cdot l$ and the complexity of the algorithm is given by $O(d^2 \cdot l^2)$. Using the FP-Tree ensures that the complexity is much less than searching through all possible combinations which is 2^{dl}.

4.3.3 Evaluation

This section evaluates the performance of the algorithm, Assisted-Aggregation in comparison to the previously modelled and studied Algorithm Info-Aggregation presented in Chapter 3. To achieve this, the number of applications $|\mathscr{A}|$, mobile devices $|\mathscr{N}|$, sensor-types $|\mathscr{H}|$ and requirements of the applications are varied.
Although, the scope of aggregating and serving sensor data to multiple applications from one device has been evaluated before, the ability to tailor reporting needs of the application has not been explored. By updating the parameters for the IoT scenario, a comparison is possible which helps to quantify the Algorithm Assisted-Aggregation. The results suggest by adapting the application-specific state machine methodology, Algorithm Assisted-Aggregation effectively further reduces the number of active mobile devices and volume available for offloading from the sensing environment whilst saving energy expended in the process.
The section is organized as follows. It begins by detailing the simulation model, illustrating experimental results and concludes by presenting the findings.

4.3.3.1 Simulation Model

Similar to the simulation setup defined in §3.5, the physical sensing area of the mobile devices is modelled as a 100m $\times$ 100m grid subdivided into request locations of size 10m $\times$

10m. These dimensions are chosen after considering the dimensions of an average house in Ireland/United Kingdom[1]. Mobile device movements within this sensing space are emulated using the Truncated Levy-Walk mobility model (Rhee et al., 2011). This model can be represented by the tuple (l, θ, t_f, t_p), where l is the flight length randomly picked up from a Levy distribution with coefficient $\alpha = 1.5$, θ is the angle of flight which follows a uniform distribution, t_f is the flight time, and t_p is the pause time which is Levy distributed with coefficient $\beta = 0.5$. The real path taken by the mobile device is defined using one instance of this model while another instance is used to predict the path taken by the mobile device for the next time-interval. This time progression is represented with a total duration of $\mathscr{T} = 240$ minutes that is divided into the discrete time intervals of duration $\Delta t = 2$ minutes. The sensor-types in a location/mobile-device are randomly selected using a uniform distribution and it is assumed that each location has at most two static sensors with the same sensor-type. Furthermore, each Application-Sensor pair defines different request locations within the grid, dependant on location constraints, that need to be covered by mobile devices. This number is limited to a maximum of four locations while the sensing time period and minimum accuracy for every pair are randomly picked from uniform distributions. Each request location contains a consolidated state machine for every sensor-type and it is assumed that the probability of offloading sensor-data for an Application-Sensor pair in a time-interval is 20%. This assumption helps in defining the reporting thresholds for the Application-Sensor pair.

For simulation purposes, BlueTooth technology is used by mobile devices to access static sensors in the location, only if the embedded sensors cannot provide sensor data with the required accuracy for an application and the collected data is pre-processed. In the Algorithm Assisted-Aggregation, the reporting device first accesses its factored state machine for that sensor-type to determine whether the data needs to be offloaded. In case, the pre-processed data is found to be beyond the reporting thresholds saved in the device, a simultaneous dialogue is initiated with the other candidate reporting devices for the location using WiFi-Direct (Pyattaev et al., 2013; Pyattaev et al., 2013). This supports one-2-one and one-2-many operations over WiFi-enabled mobile devices but does not require a WiFi access point, allowing peer-2-peer transmissions between the mobile devices.

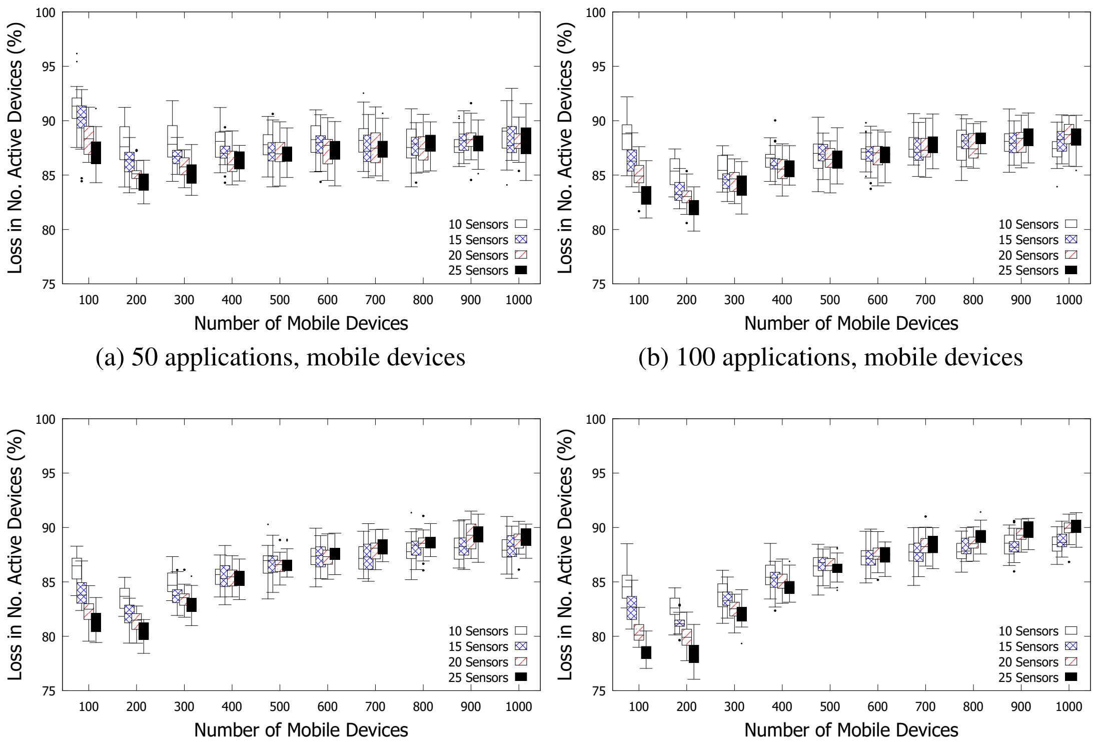
At the beginning of every simulation run, a mobile device is assigned a maximum energy of 5Wh (Apple) which decreases over time, attributed to energy consumed for general usage, for accessing the embedded sensors and static sensors, and for transmission of offloaded sensed data. Every embedded sensor-type has a specific energy consumption when accessed

[1] How Big is a House? http://shrinkthatfootprint.com/how-big-is-a-house

from the mobile-device. This is modelled using information available from Sensirion regarding environmental sensors for mobile devices with energy consumption as low as 2μW (Sensirion, a,c) and LittleRock (Priyantha et al., 2011) that presents a table on energy consumed by different sensor-types. From the range 0.002mW to 2.24mW (Priyantha et al., 2011), the energy consumed by the various sensor-types is randomly selected along with the percentage of energy consumed for general usage. Another variable associated with the sensor-type is the volume of sensed data that is collected over time. This value is randomly selected between 8 bits/second (Sensirion, b) and 50 bits/second. For energy transmission calculations, the mobile device loses 0.05W (Balasubramanian et al., 2009) to maintain WiFi connections. For accessing static sensors over BlueTooth, interacting with the candidate reporting devices using WiFi direct or transferring sensed data to the cloud, the transmission energy as presented by Friedman et al. (2013) across BlueTooth, WiFi (ad-hoc and with access-points) is availed to send/receive data. One of the communication protocols for WiFi networks(ad-hoc or access-points) as presented by Friedman et al. (2013) is randomly selected in order to cover all possible transmission channels.
Most of these parameters are similar to the simulation setup in both Chapter 3 and Chapter 5.

4.3.3.2 Results and Analysis

The rise of the Internet of Things paradigm will ensure a steady rise in the number of sensors embedded in mobile devices and the number of context-aware applications. Provisioning for this, the number of applications are varied between {50, 100, 150 and 200} that request for sensed data between {10, 15, 20 and 25} unique sensor-types. The number of mobile devices in the experiments that sense, collect, pre-process and offload this sensor data is also varied in the range of 100 to 1,000. Each run of the experiment is then identified by a fixed number of applications $|\mathscr{A}|$, mobile devices $|\mathscr{N}|$ and the maximum sensor-types in a location/mobile device $\mathscr{H}$, which is taken to be equal to the maximum number of sensor-types requested by an application. Each of these applications, mobile devices and sensor-types are identified by using a unique integer id and the experiments are run thirty times using different random number generator seeds.
The performance of the Algorithm Assisted-Aggregation in comparison with the Info-Aggregation algorithm is recorded and presented as the percentage difference of residual energy in all mobile devices, volume of data offloaded and the mean number of mobile devices that actively report the sensed data during the planning horizon in each case. Fig. 4.8 shows the mean reduction in the number of mobile device reporting sensed data to the cloud while Fig. 4.9 shows the mean percentage difference in the cumulative volume of



(a) 50 applications, mobile devices

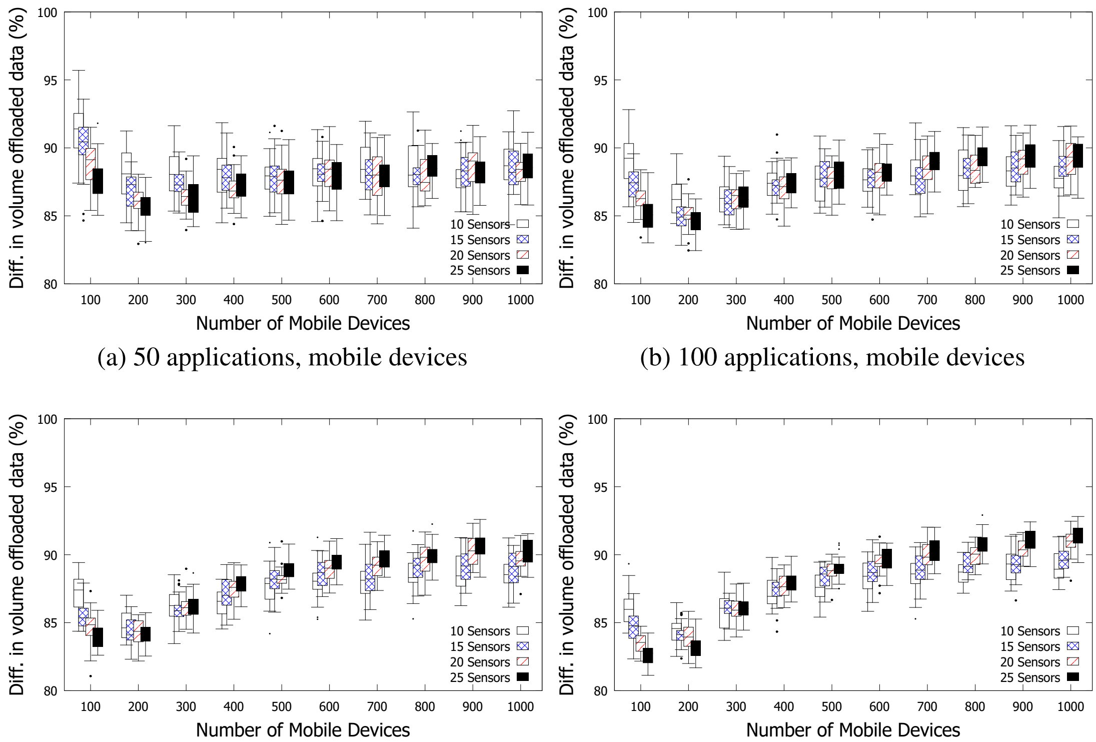
(b) 100 applications, mobile devices

(c) 150 applications, mobile devices

(d) 200 applications, mobile devices

Fig. 4.8 Comparative mean percentage difference in the number of active mobile devices offloading sensed data to the cloud between the two algorithms, Algorithm Info-Aggregation and Algorithm Assisted-Aggregation vs. the total number of mobile devices in the sensing environment for four cases using 50, 100, 150 and 200 applications. It can be seen that in all cases, Algorithm Assisted-Aggregation requires the activation of significantly fewer mobile devices.

data offloaded for the Assisted-Aggregation algorithm in comparison to the Info-Aggregation algorithm. These results show that more than 80% reduction can be achieved by aggregating sensed data from mobile devices for multiple applications and by incorporating state-machines to determine reporting needs of these applications, for both parameters. Additionally, this also highlights how the mean cumulative residual energy present in mobile devices can be saved as depicted in Fig. 4.10. This is attributed to the decreased number of reporting mobile devices as well as to the fewer message transmissions due to the use of state-machines. This effect is amplified as the number of applications increase. Additionally, it can also be observed that the cumulative energy gain decreases as the number of mobile devices in the sensing area increase. This relates to the increase in the number of mobile devices capable of covering one location. For improved global energy-efficiency, both algorithms select a larger set of active mobile devices to cover application requirements,



(a) 50 applications, mobile devices

(b) 100 applications, mobile devices

(c) 150 applications, mobile devices

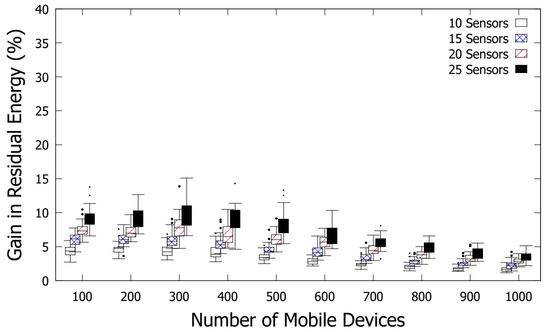
(d) 200 applications, mobile devices

Fig. 4.9 Comparative mean percentage difference in the volume of sensed data offloaded to the cloud between the two algorithms, Algorithm Info-Aggregation and Algorithm Assisted-Aggregation vs. total number of mobile devices in the sensing environment over four cases using 50, 100, 150 and 200 applications. The use of aggregation and state machines means that the Algorithm Assisted-Aggregation offloads a lower volume of data than does Algorithm Info-Aggregation.

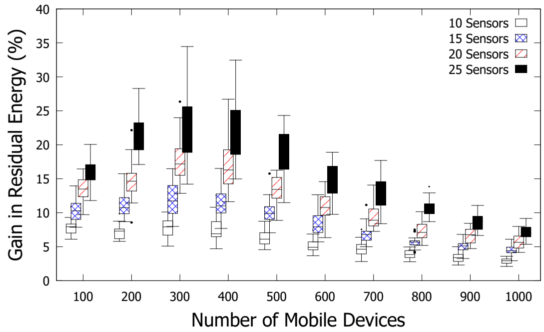
which decreases the energy gain for Algorithm Assisted-Aggregation, despite the difference in the number of active mobile devices offloading sensed data between the two algorithms. Further data analysis and model implementation for improved energy-efficiency will be done as part of future work of this dissertation.

4.4 Conclusion

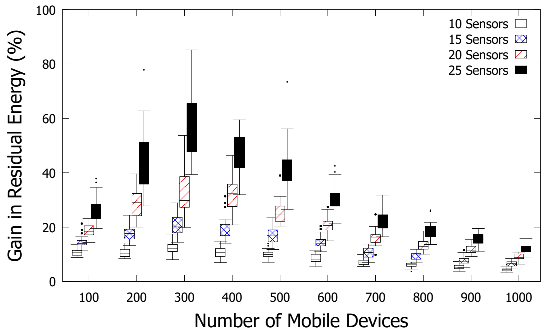
The aim of the third research question (RQ3) stated in 1.1.3 was to quantify context-awareness to determine the manner in which requirements from multiple applications can be manipulated by the framework to enable collective decision making. This was essential to efficiently offload sensed data to the application cloud by understanding



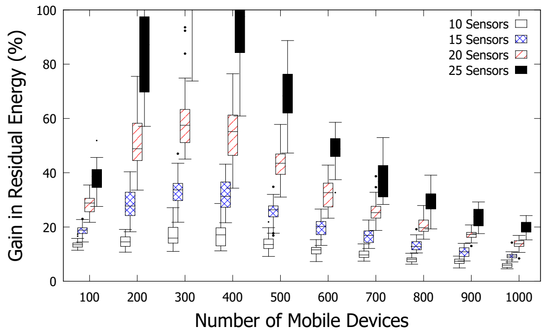
(a) 50 applications, mobile devices



(b) 100 applications, mobile devices



(c) 150 applications, mobile devices



(d) 200 applications, mobile devices

Fig. 4.10 Comparative percentage difference in the cumulative residual energy held in mobile device batteries between the two algorithms, Algorithm Info-Aggregation and Algorithm Assisted-Aggregation vs. the total number of mobile devices in the sensing environment for four cases using 50, 100, 150 and 200 applications. It is seen that that Algorithm Assisted-Aggregation results in a lower level of cumulative energy use by mobile devices, due both to the lesser reporting mobile devices and to a lower number of messages with sensed data being transmitted due to the use of aggregation.

when the data is needed. For conforming the framework to a futuristic IoT scenario, the fourth research question (RQ4) stated in 1.1.4 aimed at creating mechanisms for reliable convergence of the sensor data surrounding the user, from mobile devices to everyday appliances made ‘smarter’, because of the IoT paradigm. The sensed data acquired from these devices must also be accessed by the framework in an intelligent manner for improved accuracy and energy savings.

This chapter answers these questions with the help of two algorithms namely Algorithm Context-Localize (Algorithm 4.1) and Algorithm Assisted-Aggregation (Algorithm 4.2). Both algorithms present examples of application-specific state machines that assist mobile devices to automatically detect and deliver sensed information to concerned applications or systems.

The first section of the chapter presents the Algorithm Context-Localize that identifies and reacts to contextual change by using hierarchical clustering, modelled after the enterprise building in which sensing needs to be done. Instead of using continuous sensing, the application-specific state-driven methodology is adopted for environment-monitoring applications such as fire detection and odour detection. The four state machine tracks the system and dynamically reacts by altering the sensing and reporting rate of the mobile devices, according to the surrounding environment.
The second section presents Algorithm Assisted-Aggregation that gets seamlessly integrated with the surrounding static IoT sensors to provide accurate sensed data to multiple applications. By using frequent pattern mining, the algorithm reduces the volume of offloaded sensed data and the number of devices actively involved in this process whilst reducing energy. This is also due to the usage of consolidated state machine that maintain the context of smaller request locations for every sensor-type that could be requested.
Thus, the main contributions of this chapter are in the form of the two algorithms, Algorithm Context-Localize and Algorithm Assisted-Aggregation that successfully utilize application specific state machine for encoding user context and offload sensed data to the application cloud server according to the specified requirements in an energy-efficient manner.

Chapter 5

Cluster-Head Trajectories for Communication-Restricted Areas

5.1 Introduction

In situations where communication is limited due to unavailability of WiFi, mobile devices expend additional energy when transmitting application-specific sensed data over cellular networks to maintain the required accuracy constraints of the applications. An alternative to this approach is suspension of the reporting operations on the devices, which could negatively affect the accuracy of the system. Since, the distribution of mobile devices and the base-station placement cannot be controlled, a promising approach is optimizing the selection of mobile devices to offload this data. This is also a challenging open problem, which has been documented in this dissertation as the fifth research question (RQ5) in Chapter 1.

This chapter presents the study and design of two energy-efficient stochastic leader-selection algorithms, that use collaboration between devices in close proximity, to optimally spread the transmission energy costs. These algorithms are more evolved than the random cluster-head selection technique presented in 4.2. The chapter begins by formulating the problem mathematically and then provides a clustering approach, incorporated within the collaborative sensing framework, that exploits shorter transmit distances for communications within the cluster and requires only suitably positioned cluster-heads to offload aggregated sensed data to the application cloud server via the cellular base station. The algorithm designs are presented next which have the benefit of potentially extending the average lifetime of all mobile devices by using successive reduction of transmission distances and sharing the transmission energy cost between the devices. Experimental results based on a

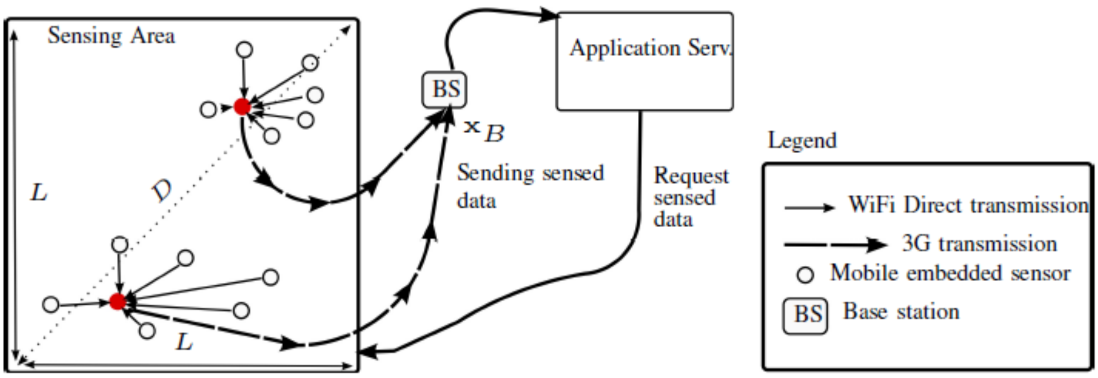


Fig. 5.1 Representation of a communication-restricted scenario for the collaborative sensing framework. Here, applications request for sensed data from mobile devices embedded with sensors, which rely on cellular networks for transmission. The sensed data is collected by one cluster-head that compresses the data before offloading it to the closest cellular base station, and it is then transferred to the application cloud server.

Java-based simulation demonstrate that the energy of the mobile devices may be increased by 20-40% without incurring a sensing accuracy penalty. The chapter then concludes by presenting a summary of the research and its applicability with respect to the research question. This work has been disseminated in Loomba et al. (2015a).

5.2 Problem Formulation

This section presents a mathematical formulation of a clustering approach to minimise energy expended during sensed data transmission to a cellular base station by mobile devices. It focuses on fairly selecting cluster-heads periodically from a given clustering solution of the mobile devices such that transmission costs are not incurred by one (a few) mobile device(s). The cluster-head trajectory minimisation formulation is presented towards the end of this section. Fig. 5.1 represents the scenario considered and Table 4.2 lists all notations used for the formulation.

The deployment of the CH selection technique in the framework is considered for open areas like parks or railway stations, in contrast to the scenarios defined in Chapter 3 and Chapter 4 which assume easy-accessibility of WiFi. Most of the notations used in formulation are however kept consistent with §3.3 and §4.3.1. This scenario can also be extended into other IoT inspired scenarios that benefit from device-to-device communication, which will be studied as part of future work. This includes modelling and simulation of clustering approaches with vehicle-to-vehicle and vehicle-to-infrastructure communication for multiple applications requesting for efficient low-latency data collection from autonomous cars.

For the mobile devices scenario, the flow of the problem starts in real-time with the application cloud server requesting for sensed data via the centralized framework from the application client installed on the mobile devices with a particular accuracy constraint. Instead of each device sending data, mobile devices collaborate amongst themselves to send sensing information about the temperature, humidity, percentage of gas etc. in the area of interest to the application. A mobile device transfers sensed information using WiFi-Direct (Pyattaev et al., 2013; Pyattaev et al., 2013) to a cluster-head who is responsible for compressing the data. WiFi-Direct supports one-2-one and one-2-many operations over WiFi-enabled devices and does not require a WiFi access point. This allows peer-2-peer transmissions between the mobile devices.

It is assumed that the participating mobile devices have location information (using GPS) and knowledge of the location of the cellular base station. This assumption is reasonable for android operating systems. Details like cell id, location area code, Mobile Country Code, Mobile Network Code are available to applications (Android Developers). In addition, several public databases like OpenCell (ENAikoon) contain a database of all base stations and are used by applications today. Thus, each cluster-head connects to one cellular base station in its micro cellular cell and offloads the sensed data.

5.2.1 Terminology

The problem is formulated over a total time duration of $\mathscr{T}$ seconds, subdivided into time intervals of Δt seconds, over which the sensed data is reported. For $t \in \mathscr{T}$ and $k \in [1, \mathscr{T}/\Delta t]$, these time intervals are denoted by $(k-1)\Delta t < t < k\Delta t$.

A mobile device $n \in \mathscr{N}$ is treated as the sensor node with equal transmission, reception and processing capabilities. $|\mathscr{N}|$ denotes the total number of mobile devices located within the area of interest which is physically assumed to be a square grid with length L. For notational simplicity, 2-D coordinates are used as opposed to 3-D and the position coordinates of a device n is given by $\mathbf{x}_n(k)$, defined as a function of the time interval k. Distance between two mobile devices, indexed by n_1 and n_2, can then be calculated in the Euclidean space as below:

$$d(\mathbf{x}_{n_1}(k), \mathbf{x}_{n_2}(k)) = ||\mathbf{x}_{n_1}(k) - \mathbf{x}_{n_2}(k)||^2. \tag{5.1}$$

Table 5.1 Notation used for Problem Formulation in Chapter 5.

Notation	Description
$\mathscr{T}$	Planning horizon
t	Time variable
Δt	Time Interval for reporting
k	Index of time interval
L	Length of Sensing area
$\mathscr{N}$	Set of mobile devices
n	Index of mobile device
$\mathbf{x}_{n(k)}$	Vector coordinates of the mobile device n in time interval k
$d(a,b)$	Euclidean distance between vector coordinates a and b
i,h	Index of a cluster
X	Total number of clusters
$\mathscr{C}_i$	Cluster at index i
$\mathbf{x}_B$	Vector coordinates of the cellular base station
V_n	Sensed data in bits for mobile device n
V_i	Sensed data available for cluster-head of cluster i after aggregation
p	Percentage of aggregation applied by the cluster-head
S_n^k	Energy incurred by mobile device n during sensing in time interval k
T_n^k	Energy incurred by mobile device n during transmission in time interval k
R_n^k	Energy incurred by mobile device n for receiving data in time interval k
E_n^k	Total energy consumed by mobile device n in time interval k
λ	Path loss exponent
y_n^k	0-1 decision variable indicating whether mobile device n is a cluster-head in time interval k
B_n	Full batter level of mobile device n
b_n^k	Battery level of mobile device n at the start of time interval k
ζ	Minimum percentage of battery that must be available in any mobile device

The mobile devices are partitioned into X clusters. Once the cluster-heads have been selected, the mobile devices select the nearest cluster-head as their cluster-head, and form the set $\mathscr{C}_i$, which is the set of members of the i-th cluster. The sets $\mathscr{C}_i, \forall i$ have the following properties

- There are no empty clusters.
- The cluster membership does not overlap.
- All mobile devices are assigned to at least one cluster.

Formally, this is given by Eq. 5.4:

$$\mathscr{C}_i \neq \emptyset, \qquad i \in [1, X] \tag{5.2}$$

$$\mathscr{C}_h \cap \mathscr{C}_i = \emptyset, \qquad h, i \in [1, X], i \neq h \tag{5.3}$$

$$\bigcup_{i \in [1,X]} \mathscr{C}_i = \mathscr{N} \tag{5.4}$$

The cluster-head for each cluster is responsible for aggregating the sensed data received from the cluster members. The sensed data available in mobile device n is assumed to be V_n bits. Then the data available for transmission after aggregation in the cluster-head is defined as:

$$V_i = p \cdot \sum_{\forall n \in \mathscr{C}_i} V_n \tag{5.5}$$

It is assumed that the aggregation factor is on average p and that the accuracy of the technique is not significantly affected by different cluster-head selections. Every Δt seconds, the data V_i, collected and aggregated from cluster $\mathscr{C}_i$, is transferred to the base station which lies at coordinates $\mathbf{x_B}$. It is assumed to typically be at a greater distance from all devices than the distance between the devices that can directly communicate with each other. This assumption is fundamental and the problem is formulated to ensure that mobile-2-mobile communications are cheaper, in terms of energy usage than mobile-2-base station communications, as given by Eq. 5.6.

$$d(\mathbf{x}_B, \mathbf{x}_n(k)) > d(\mathbf{x}_n(k), \mathbf{x}_j(k)), \forall j, n \in \mathscr{N}, \forall k \tag{5.6}$$

The energy consumed by mobile n in time interval k is due to its support of three functionalities

- Sensing, which costs S_n^k
- Transmission of data, which costs T_n^k
- Reception of data, either mobile-2-base station or mobile-2-mobile, which costs R_n^k.

The dominant term in this sum is T_n^k and the total energy expended is approximately equal to the transmission cost. Also, the energy lost during mobile-2-base station transmissions is

proportional to the distance $d(\mathbf{x}_B, \mathbf{x}_n(k))$ raised to the power of λ. Here λ denotes the path loss exponent for cellular transmissions, e.g. mobile-2-base station transmissions.
Thus total energy expended by mobile n during interval k is shown in Eq. 5.7 and for any cluster-head n actively reporting to the base station, Eq. 5.8 applies. Instead of using $T_n^k = a + b \cdot d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda$, the proportionality factors and offsets are dropped. Thus $T_n^k = d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda$ by assigning $a = 0$ and $b = 1$. This is done to simplify the notation as presented in this chapter.

$$E_n^k = S_n^k + T_n^k + R_n^k \tag{5.7}$$

$$E_n^k \approx T_n^k \propto d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda \tag{5.8}$$

5.2.2 Problem Statement

Given the terminology and assumptions above, the optimization problem for the efficient selection of cluster-heads is presented below.

5.2.2.1 Decision Variables

The decision variable $y_n^k \in \{0, 1\}$ indicates whether the mobile devices becomes a cluster-head during time interval k. If $y_n^k = 1$, mobile device n is a cluster-head for time interval k.

5.2.2.2 Constraints

The battery life constraint is considered for this formulation. As defined earlier in Chapters 3 and Chapter 4, an important constraint is the restriction on the battery level of each device to ensure one device dying does not affect the sensing solution. Thus if the battery level of mobile device n at the start of time interval k, denoted b_n^k, is below ζ of the full battery level B_n, the mobile device is powered off. This restriction is imposed on all mobile devices for every time interval to ensure battery availability for sensing, transmission and receiving and is presented as Eq. 5.9.

$$\begin{aligned} &\forall n \in \mathscr{N}, \forall k \in [0, \mathscr{T}/\Delta t] : \\ &b_n^k >= \zeta B_n \\ &b_n^{k+1} = b_n^k - S_n^k - T_n^k - R_n^k \end{aligned} \tag{5.9}$$

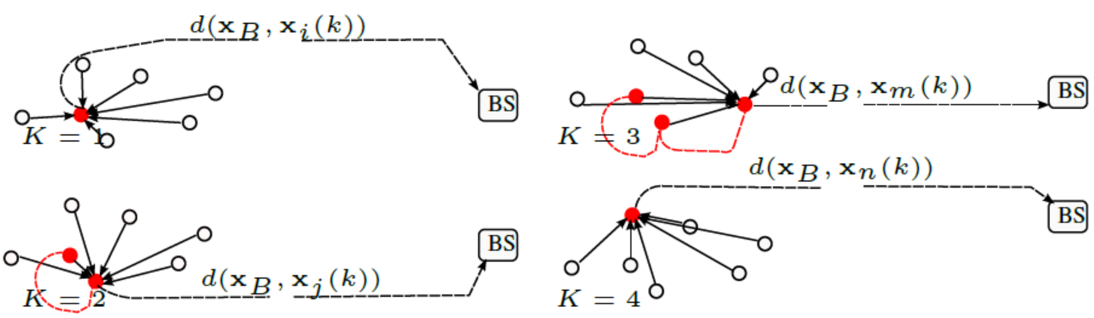


Fig. 5.2 Representation of the cluster-head trajectories at time interval k = 1, 2, 3 and 4, where cluster-heads are shown in red dots and the transmissions between the mobile devices and the cluster-head is shown by solid black arrows. The transmission between the cluster-head at each time to the base station is illustrated with a dashed black arrow, and $d(\mathbf{x}_B, \mathbf{x}_n(k))$ meters. The inter-mobile distance is small relative to the cluster-head-2-base station distance. For time interval k = 1, 2, 3, the cluster-head follows a trajectory (dashed red line), which moves the cluster-head successively closer to the base station, reducing the cost of the cluster-head-2-base station transmission. At time interval $k = 4$ the cluster-head jumps to a new location. For time interval k = 1, 2, 3, energy is saved. At time interval $k = 4$, less energy is saved, m_n is smaller, but fairness is preserved.

5.2.2.3 Objective Function

The property that is exploited to decrease energy usage is described as follows. Referring to (Eq. 5.6) the crucial point is that the marginal cost of the two types of transmission, mobile-2-mobile and mobile-2-base station, grows rapidly as a function of the transmission distances, raised to the exponent λ. Thus the energy saved when mobile device n performs a local transmission to mobile device j as opposed to a transmission to the base station can be quantified by m_n as given in Eq. 5.10. The greater the average distance of the mobile devices from the base station, $\frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} d(\mathbf{x}_B, \mathbf{x}_n(k))$, the greater the gain in energy saving can be expected.

$$m_n = |d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda - d(\mathbf{x}_n(k), \mathbf{x}_j(k))^\lambda| \tag{5.10}$$

For analysis, a simple case with one cluster $X = 1$ is considered for the best and worst case scenarios. In the worst case scenario, all mobile devices will communicate directly with the base station at a cost $\sum_{n\in\mathcal{N}} d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda$. In the best case scenario, all mobile devices communicate with one cluster-head, j, who transmits all of the sensed data to the base station, at a cost $d(\mathbf{x}_B, \mathbf{x}_j(k))^\lambda$. The total saving for each mobile device that does not communicate with the base station is $\sum_{n\in\mathcal{N}\setminus j} m_n$. Let

$s(j) = d(\mathbf{x}_B, \mathbf{x}_j(k))^\lambda + \sum_{n \in \mathscr{N} \setminus j} d(\mathbf{x}_n(k), \mathbf{x}_j(k))^\lambda$. Using these values, it can be seen that the lower bound will be 0 and the upper bound u can be constructed from the energy cost of the worst case, where all mobile devices transmit to the base station, and the alternative, which is where the best mobile device is chosen to be the cluster-head. This is given as Eq.5.11.

$$u = \sum_{n \in \mathscr{N}} d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda - \min_j s(j) \tag{5.11}$$

This helps in getting a good estimate of the potential energy saving. To ensure that Eq. 5.6 holds, it is assured that the set of mobile devices that lie within the distance d, a user defined constant, of cluster-head, n, is denoted $\mathscr{W}_n$. This bounds the maximum distance of mobile-2-mobile transmissions so that energy savings are achieved with high probability. A less exact, simplified upper bound follows

$$\sum_{n \in \mathscr{N}} d(\mathbf{x}_B, \mathbf{x}_n(k))^\lambda - \min_j \left(|\mathscr{N} \setminus j| d^\lambda + d(\mathbf{x}_B, \mathbf{x}_j(k))^\lambda \right) \tag{5.12}$$

Taking this analysis one step further, and considering the simple case of $X = 1$ cluster-heads, a greedy mechanism for reducing energy consumption, if the location of the mobile devices is known, is to consider the formation of chains of cluster-heads through time, where each successively chosen cluster-head reduces the energy cost of transmission to the base station, e.g. by selecting mobile devices i, j, and m and so on. Thus the objective function of this chapter is to minimise the cluster-head trajectory which can be formalized as:

$$d(\mathbf{x}_B, \mathbf{x}_i(k)) > d(\mathbf{x}_B, \mathbf{x}_j(k+1)) > d(\mathbf{x}_B, \mathbf{x}_m(k+2)) \ldots \tag{5.13}$$

This procedure is illustrated in Fig. 5.2. The cluster-head is moved closer to the base station as k increases, which is far away from the mobile devices; this movement is called the *trajectory* of the cluster-heads and it is illustrated in red. It does not involve the physical movement of any of the mobile devices, but a change in the role that they perform.
This adaptive cluster-head trajectory concept is also inspired from natural coordination behaviour and leadership roles in biological systems. It is particularly influenced by the complex human immunological response against invasion and spread of invading pathogens or antigens. The white blood cells/lymphocytes that contain antibodies for various antigens are called B-cells. On identification of a foreign presence, these get stimulated and secrete their antibodies towards the direction of the identified invasion. The level of B-cell stimulation depends on its distance from the invading pathogen and its correlation with the surrounding B-cells. This creates a series of immunological synapses or a synaptic relay race

Algorithm 5.1 Algorithm CH-Trajectory

Require: Probability p, Time Interval k, constant D

for all $n \in \mathscr{N}$ **do**
    $\mathscr{W}_i$= GETNEARESTNEIGHBOURS(CHi,$k-1$)
    B= GETBASE(p,k)
    $f(\mathbf{x}_B, \mathbf{x}_{n(k)}) = \frac{1}{2}\left(\frac{D-d(\mathbf{x}_B,\mathbf{x}_{n(k)})}{DB} + 1\right)$
    $S = \max f(\mathbf{x}_B, \mathbf{x}_{n(k)})$
    if $n \in \mathscr{W}_i$ **then**
        $D_n^k = \frac{B}{S} \cdot f(\mathbf{x}_B, \mathbf{x}_{n(k)})$
    else
        $D_n^k = B$
    end if
    r=GETRANDOMNUMBER([0,1])
    if $r < D_n^k$ **then**
        isCluster-Head=true;
        SENDSELFELECTIONMSG
        WAITFORACK
        SENDTRANSMISSIONSCHEDULE
        RECEIVESENSEDDATA
        SENDAGGDATATOBASE STATION
    else
        isCluster-Head=false;
        WAITFORELECTIONMSG
        SELECTCLOSESTCLUSTER-HEAD
        SENDACKTOCLUSTER-HEAD
        WAITFORTRANSMISSIONSCHEDULE
        TRANSMITSENSEDDATA
    end if

Ensure: CHs for Time Interval k

end for

to proliferate and protect the body without centralized brain control. This is similar to the cluster-head trajectory minimisation approach presented in this chapter.
In the next section, two stochastic algorithms are presented that follow this cluster-head trajectory minimisation approach for energy savings.

5.3 Algorithm Design

This section describes two greedy algorithms, Algorithm CH-Trajectory (Algorithm 5.1) and Algorithm CH-Trajectory-With-Forgetting-Factor (Algorithm 5.2) that stochastically select

Algorithm 5.2 Algorithm CH-Trajectory-With-Forgetting-Factor

Require: Probability p, Time Interval k, constant D, $l = 0.8$

for all $n \in \mathcal{N}$ **do**
  $\mathscr{W}_i$= GETNEARESTNEIGHBOURS(CH i,$k-1$)
  B= GETBASE(p,k)
  $f(\mathbf{x}_B, \mathbf{x}_{n(k)}) = \frac{1}{2}\left(\frac{D-d(\mathbf{x}_B,\mathbf{x}_{n(k)})}{DB} + 1\right)$
  v = SETINTERVALADDED
  $g(k) = \left(\frac{1}{(k-v+1)^l}\right)$
  if $n \in \mathscr{W}_i$ & $B < B \cdot f(\mathbf{x}_B, \mathbf{x}_{n(k)}) \cdot g(k)$ **then**
    $D_n^k = B \cdot f(\mathbf{x}_B, \mathbf{x}_{n(k)}) \cdot g(k)$
  else if $B > B \cdot f(\mathbf{x}_B, \mathbf{x}_{n(k)}) \cdot g(k)$ **then**
    REMOVEMODULATION
  else
    $D_n^k = B$
  end if
  r=GETRANDOMNUMBER([0,1])
  if $r < D_n^k$ **then**
    isCluster-Head=true;
    SENDSELFELECTIONMSG
    WAITFORACK
    SENDTRANSMISSIONSCHEDULE
    RECEIVESENSEDDATA
    SENDAGGDATATOBASE STATION
  else
    isCluster-Head=false;
    WAITFORELECTIONMSG
    SELECTCLOSESTCLUSTER-HEAD
    SENDACKTOCLUSTER-HEAD
    WAITFORTRANSMISSIONSCHEDULE
    TRANSMITSENSEDDATA
  end if

Ensure: CHs for Time Interval k

end for

the cluster-head for the next time interval and optimize the energy usage of each current cluster-head in a collaborative manner. In both algorithms, the transmission costs is distributed among the mobile devices by adopting the cluster-head trajectory minimisation technique illustrated in Fig. 5.2.

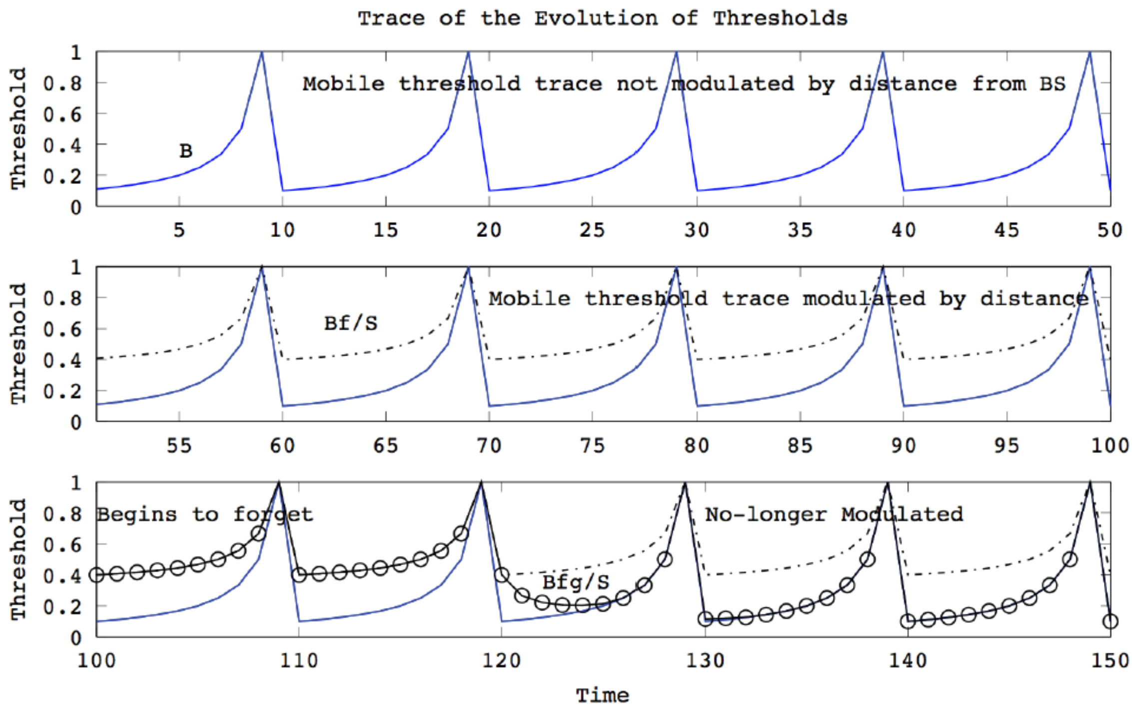


Fig. 5.3 Illustration of the threshold modulation by Algorithm CH-Trajectory and Algorithm CH-Trajectory-With-Forgetting-Factor in comparison to the LEACH algorithm. The first row (solid line) shows the base threshold B trace, for intervals $1 \leq k \leq 50$ (Eq. 5.15). At time $51 \leq k \leq 100$ the distance function defined in Algorithm CH-Trajectory is incorporated into the trace (Eq. 5.17), which is shown in the second row (dashed-dot line). The minimum value of the threshold is thus modulated by the distance of the mobile from the base station. In the third row (o-solid line), for intervals $101 \leq k \leq 150$, the forgetting factor defined in Algorithm CH-Trajectory-With-Forgetting-Factor (Eq. 5.19) is incorporated and removes the effect of the distance function from the threshold by time interval $k = 130$.

5.3.1 Threshold Definition

A threshold value, D_n^k is defined for each mobile device n for time interval k and is used to determine the probability that the device n becomes a cluster-head. This is done by random number comparison. Each mobile device randomly draws a number between 0 and 1 and compares the outcome of the trial, r, with the threshold, D_n^k. Then the decision rule for determining the cluster-head-ship of a mobile device n for time interval k is given in

Eq. 5.14. Thus the threshold D_n^k is a periodic function and determines the expected number of cluster-heads during time interval k. The

$$y_n^k = \begin{cases} 0, & \text{if } r \geq D_n^k, \\ 1, & \text{if } r < D_n^k, \end{cases} \tag{5.14}$$

Similar to the LEACH algorithm (Heinzelman et al., 2000), a cluster-based technique widely used for wireless sensor networks, a *base threshold*, B is defined as a function of time in Eq. 5.15.The parameter p denotes the probability of a mobile device to be a cluster-head. Since, p is the same for all mobile devices, each mobile traces out exactly the same threshold evolution. This is also visualized in Fig. 5.3, row 1, which depicts the evolution of five iterations of the base threshold. Here, the parameter p sets the minimum value this cyclic function achieves. Thus the larger the value of p, the greater the number of mobile devices that will be selected as cluster-head.

$$D_n^k = B = \frac{p}{1 - p \bmod (k, \frac{1}{p})}, \forall n \in \mathscr{N}, \forall k \tag{5.15}$$

5.3.2 Threshold Modulation

This base threshold is then modulated for some mobile devices in both Algorithm CH-Trajectory and Algorithm CH-Trajectory-With-Forgetting-Factor by incorporating location information which is the main difference to the LEACH algorithm. Additionally, Algorithm CH-Trajectory-With-Forgetting-Factor also includes a mechanism for adapting and forgetting the threshold modulation based on the mobility of the mobile devices. Furthermore, both algorithms support multiple-selection of a device to become a cluster-head and the number of cluster-heads X can also change with time. This section describes the threshold modulation techniques and presents the pseudo-code of both algorithms. The effects of modulating the threshold in the different algorithms is depicted in Fig. 5.3.

5.3.2.1 For Algorithm CH-Trajectory

The pseudo-code for the algorithm is presented as Algorithm 5.1 and it alters the threshold for the nearest neighbours $\mathscr{W}_i$ of the cluster-head, i, by a function which depends on their distance to the base station as given from line 2 to 10.

The threshold of mobile device $n \in \mathscr{W}_i$ for the decision rule (Eq. 5.14) is modulated by including a *distance function* factor as given in Eq. 5.16.

$$f(\mathbf{x}_B, \mathbf{x}_n(k)) = \frac{1}{2}\left(\frac{D - d(\mathbf{x}_B, \mathbf{x}_n(k))}{DB} + 1\right) \tag{5.16}$$

Here, the constant D defines the maximum permissible mobile-2-mobile communication distance which is a user defined value. A good choice for this constant for Fig. 5.1 is $D = \sqrt{2L^2}$. The term, $S = \max f(\mathbf{x}_B, \mathbf{x}_n(k)) \cdot B$, is the maximum value that the distance function, f can achieve for a given position $\mathbf{x}_n(k)$. Every time the mobile device moves, the values of f and S are updated. Thus the modulated threshold is defined as below:

$$D_n^k = \begin{cases} \frac{B}{2S}\left(\frac{D - d(\mathbf{x}_B, \mathbf{x}_n(k))}{DB} + 1\right) & \forall n \in \mathscr{W}_i \\ B & \forall n \in \mathscr{N} \setminus \mathscr{W}_i \end{cases} \tag{5.17}$$

The advantage of including $\frac{f}{S}$ is that greater energy is saved with high probability as cluster-head trajectories, like the trajectory in Eq. 5.13, are followed. However, the problem with such an approach is that when the cluster-head has moved as close as possible to the base station, it cannot move closer, and the mobile device may become a cluster-head too frequently. This means that the algorithm is not as fair as the LEACH algorithmic approach (Eq. 5.15) and the battery of such devices will continuously be depleted as they are unfairly sending the costly offload transmissions to the base station.

5.3.2.2 For Algorithm CH-Trajectory-With-Forgetting-Factor

For improved fairness, the second algorithm is presented as Algorithm 5.2 that stochastically selects the cluster-head in such a way that the cluster-head-ship may jump out of its current trajectory and start a new trajectory at a new position. This process is illustrated in Fig. 5.2, row 3 at time interval $k = 4$. A new mobile device is selected as the cluster-head, which has no relation with previous cluster-heads. This jump disperses the costly transmissions to the base station between the mobile devices, introducing fairness into the cluster-head selection.

This is achieved by incorporating an additional forgetting factor g as defined in Eq. 5.18 into the threshold trace generation function in Eq. 5.17, where v denotes the time when mobile device n starts modulating its threshold by f/S. This is presented in the pseudo code of the algorithm from line 2 to 12. The power in the denominator l sets the rate at which the mobile forgets.

$$g(k) = \left(\frac{1}{(k-v+1)^l} \right) \text{when } n \in \mathscr{W}_i \tag{5.18}$$

In Fig. 5.3, l is defined to be equal to 0.8 and $v = 100$, which implies that a neighbouring mobile was elected cluster-head at time $k = 99$. The choice $l = 0.8$ means the effect of f is forgotten after 30 samples. Multiplying g, a decay function by the modulation factor f causes the effect of the modulation factor to be forgotten.

This forgetting function is incorporated for each mobile device that was in the neighbourhood of the cluster-head, i, e.g. the set $\mathscr{W}_i$ in an previous interval, but that was not selected to be a cluster-head. The role of g is to ensure that after a few intervals of not being chosen to be a cluster-head, mobile device n stops modulating its threshold. So, once this new threshold value becomes less than the base threshold, the mobile no longer modulates the threshold, unless the mobile is in the neighbourhood of a cluster-head at some future time. This ensures that the mobile devices forgets about location information, if it may be outdated, and about previous cluster-head assignments so that after a suitable time has elapsed the mobile device uses the fair threshold generating rule (in Eq. 5.15). The modulated threshold is then defined as below:

$$D_n^k = \begin{cases} B \cdot f(\mathbf{x}_B, \mathbf{x}_n(k)) \cdot g(k), & \forall n \in \mathscr{W}_i \\ \text{if } B \leq B \cdot f(\mathbf{x}_B, \mathbf{x}_n(k)) \cdot g(k). & \text{otherwise} \\ B & \forall n \in \mathscr{N} \setminus \mathscr{W}_i \end{cases} \tag{5.19}$$

5.3.3 Control of Flow

The algorithms are divided into three phases in Fig. 5.4 namely Initial, Cluster-Setup and Transmit to select each new cluster-head every Δt seconds.

5.3.3.1 Initial Phase:

In this phase, the cluster-heads for the interval are selected in a distributed way. Each mobile selects a uniformly distributed random variable r in the range [0,1] using the function getRandomNumber() in line 11 for Algorithm 5.1 and line 14 for Algorithm 5.2. If this number is less than the threshold defined for the interval, D_n^k, the mobile becomes a cluster-head for that time interval (Eq. 5.14). The base-threshold is used to initiate the

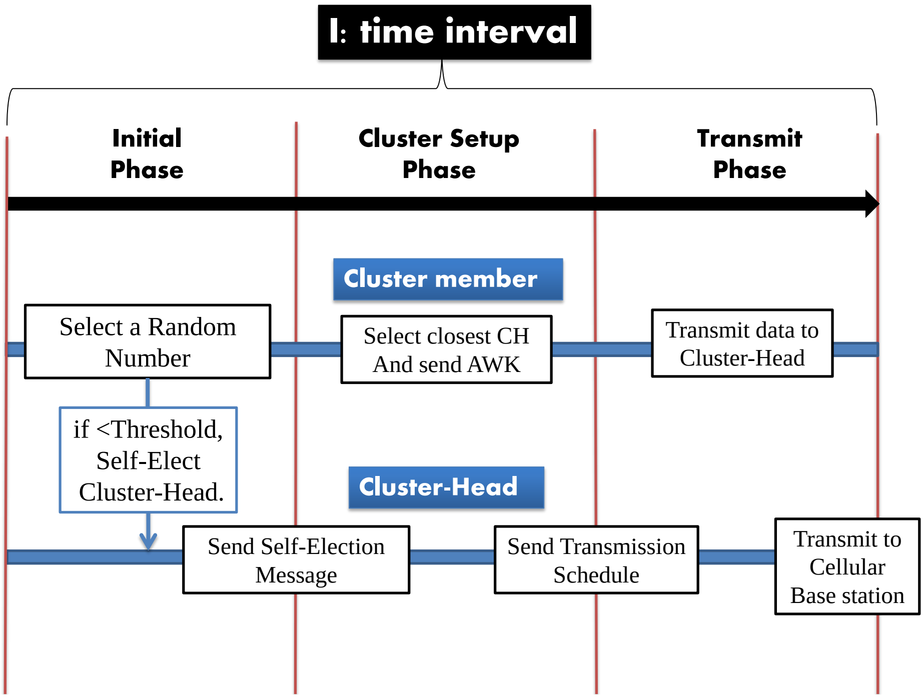
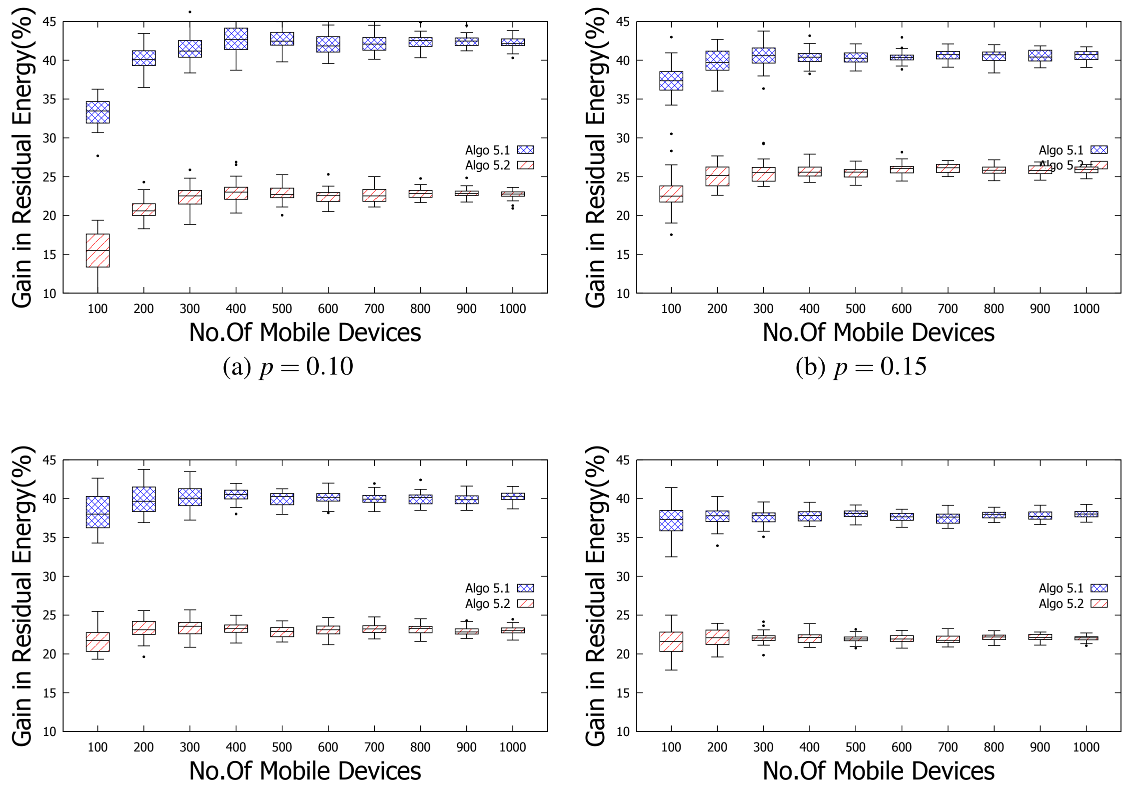


Fig. 5.4 Representation of the three phases of both the clustering algorithms, within a time interval k. Each interval Δt consists of an initial, cluster set-up and transmit phase. Cluster-heads are selected based on a random trial in the initial phase. The cluster-head announces its selection and the other mobile devices select the closest cluster-head as their cluster-head during the cluster set-up phase. The cluster-head sends a transmit schedule and the mobile devices transmit sensing data to the cluster-head during the transmit phase. Finally, the cluster-head transmits the aggregated data to the base station.

thresholds during the first interval, $k = 1$. If mobile n is a cluster-head during interval $k > 1$, the thresholds of the members of the set $\mathscr{W}_n$ are modulated by the distance function f/S if Algorithm 5.2 is used, or the distance function and the forgetting factor f/S and g if Algorithm 5.2 is used. Therefore, the outcome of future stochastic trials is biased by the identity of the cluster-head during the previous interval n and the distance of the mobile devices in the set $\mathscr{W}_n$ from the base station. Mobiles that are not members of the $\mathscr{W}_n$, or the neighbour sets of other cluster-heads use the base-threshold B to perform cluster-head selection trials.

5.3.3.2 Cluster Setup Phase:

Each non-cluster-head mobile must choose to belong to one cluster-head. Due to the mobility of the mobile devices, the cluster-head chosen is the closest (using Eq. 5.1) cluster-head during that time-interval. Each mobile predicts its mobility path in the time-interval k and



(a) $p = 0.10$ (b) $p = 0.15$

(c) $p = 0.20$ (d) $p = 0.25$

Fig. 5.5 Mean percentage gain in the cumulative residual energy held in mobile device batteries for the two algorithms, Algorithm CH-Trajectory and Algorithm CH-Trajectory-With-Forgetting-Factor for Algorithm in comparison to the LEACH algorithm vs. the total number of mobile devices in the sensing environment for four different values of the probability of a device becoming a cluster-head, $p \in \{.1, .15, .2, .25\}$.

the cluster-head with the closest cluster-head is selected for that time-interval. An acknowledgement is then sent to the selected cluster-head to inform it about its new member.

5.3.3.3 Transmit Phase:

The cluster-head receives all mobile messages and creates a transmission schedule for all of its member mobile devices. The members transmit the sensed data during their allocated time. When the data is received, the cluster-head compresses the data into a single data stream and transmits it to the base station.

Table 5.2 Mean Percentage Gain (Standard Deviation in Braces) between Algorithm CH-Trajectory-With-Forgetting-Factor in comparison to the LEACH algorithm for different total number of mobile devices in the sensing environment and different forgetting-factor slopes.

	Total Mobile Devices	$p = .1$	$p = .15$	$p = .2$	$p = .25$
$l = .6$	100	16.2(2.15)	23.3(2.17)	22.1(1.68)	21.9(1.61)
	200	22.3(1.48)	25.5(1.24)	23.4(1.44)	22.4(1.04)
	300	23.0(1.42)	26.4(1.26)	23.6(1.15)	22.3(0.74)
	400	23.9(1.33)	25.9(0.76)	23.8(0.73)	22.3(0.61)
	500	23.2(1.28)	25.9(0.9)	23.4(0.57)	22.2(0.55)
	600	23.1(1.06)	26.3(0.76)	23.5(0.76)	22.3(0.54)
	700	23.4(0.82)	26.4(0.6)	23.7(0.63)	22.3(0.62)
	800	23.5(0.74)	26.3(0.66)	23.5(0.68)	22.4(0.49)
	900	23.6(0.59)	26.3(0.68)	23.5(0.53)	22.4(0.44)
	1000	23.3(0.68)	26.2(0.55)	23.6(0.48)	22.4(0.52)
$l = .9$	100	11.1(2.13)	18.3(2.17)	16.3(1.51)	16.2(1.42)
	200	15.6(1.28)	19.7(1.33)	17.4(0.94)	16.2(1)
	300	16.7(1.26)	20.4(1.32)	17.6(1.03)	16.5(0.71)
	400	17.1(1.37)	20.3(0.69)	17.8(0.65)	16.5(0.64)
	500	17.3(0.9)	20.3(0.81)	17.2(0.75)	16.2(0.52)
	600	16.8(0.98)	20.5(0.83)	17.6(0.67)	16.4(0.48)
	700	17.2(0.68)	20.9(0.74)	17.9(0.67)	16.6(0.52)
	800	17.3(0.7)	20.6(0.54)	17.8(0.62)	16.6(0.41)
	900	17.1(0.51)	20.6(0.54)	17.5(0.54)	16.4(0.53)
	1000	16.9(0.54)	20.5(0.5)	17.5(0.5)	16.3(0.42)

5.4 Evaluation

This section evaluates the performance of Algorithm 5.1 and Algorithm 5.2 in comparison to the LEACH algorithm, to calculate the distribution of energy savings for all mobile devices that are involved in this process. To achieve this, the total number of mobile devices, the probability p of each mobile device becoming a cluster-head, and the slope of the forgetting factor l are varied. The results suggest that by adapting the cluster-head trajectory methodology, both algorithms are able to reduce the energy expended by the mobile devices The section is organized as follows. It begins by detailing the simulation model, illustrating experimental results and concludes by presenting the findings.

5.4.1 Simulation Model

The physical sensing area is modelled using a square grid (100m $\times$ 100m). This represents the simulation study area and bounds the trajectory of each mobile device. The Truncated

Levy-Walk mobility model (Rhee et al., 2011), represented by the tuple (l, θ, t_f, t_p) is used to determine the path of a mobile device. As defined in the previous chapters, l is the flight length randomly picked up from a Levy distribution with coefficient $\alpha = 1.5$, θ is the angle of flight which follows a uniform distribution, t_f is the flight time calculated using a constant speed of 1m/s during the flight, and t_p is the pause time which is Levy distributed with coefficient $\beta = 0.5$. The scale parameters of the flight length and pause times in the Levy distributions are selected as 0.01m and 1000s and the truncation factors are defined as 100m and 1000s respectively.

Time progression within the simulation study is modelled in discrete time steps of duration $\Delta t = 2$ minutes and for all the experiments described below, $\mathscr{T} = 240$ minutes is the overall simulated duration. The position of the mobile device is updated every 40s, three times per time-interval Δt. These values are motivated by the fact that the mobile devices are within a mean value of one metre from their original position when a time interval elapses.

The mobile devices have a maximum energy of 5Wh (Apple) at the beginning of the simulation which decreases over time due to the general usage of the mobile, the energy consumed for the operation of the sensors and the energy used for transmitting sensed data. Sensirion offers humidity and temperature sensors for mobile devices with an energy consumption of 0.01152 J/h (Sensirion, c). This is taken to be the energy loss associated with sensing.

The cellular base station is assumed to be at location $\mathbf{x}_B = [500, 500]^T$. For energy transmission calculations, the COST-231 propagation model based on the Walfish-Ikegami model for micro-cell deployments (Damosso and COST Telecom Secretariat, 1999; Sarkar et al., 2003) is adopted. Additionally, the non-line-of-sight path loss model for 2GHz is used which is $35.7 + 38log_{10}(d)$. The mobile device also uses energy to maintain connections, 0.02 J/sec for the cellular network(3G) and 0.05 J/sec for WiFi (Balasubramanian et al., 2009). This is used as maintenance energy cost for WiFi Direct communications.

Most of these parameters are similar to the simulation setup in both Chapter 3 and Chapter 4.

5.4.2 Results and Analysis

Each simulation run is specified by the number of mobile devices and probability of a device becoming a cluster-head. The total number of mobile devices $|\mathscr{N}|$ sensing the environment are varied in steps of 100 from 100 to 1000 and the probability p of a cluster-head being selected is examined to study how different probabilities $p \in \{.1, .15, .2, .25\}$ affect energy savings. Each experiment is also randomly initialized and run thirty times. Fig. 5.5 shows the comparison of the two algorithms with LEACH and box-plots are computed for each pair of p and $|\mathcal{N}|$.

It can be concluded that both the cluster-head trajectory minimisation algorithms improve the residual energy of the mobile device over the LEACH algorithm (by $\approx 40\%$ and $\approx 20\%$) irrespective of the number of mobile devices $|\mathcal{N}|$ and the probability p. Also, as expected, the variation in the energy gains decreases as the number of mobile devices increases. In terms of the role of the probability p in determining the percentage energy saving, the gain in energy saving over LEACH decreases as the probability of each mobile being a cluster-head increases. Furthermore, Algorithm 5.1 achieves better gains than Algorithm 5.2 but the number of mobile devices used as cluster-heads is greater for Algorithm 5.1. This is explained by considering the role of the slope of the forgetting factor l. It can also be seen that both algorithms are fair, in that remarkably, the inter-quartile range of the energy saving (over LEACH) is $\approx 1\%$. This implies that most devices achieve an energy saving which is within 1 or 2 % of the mean energy saving.

The effect of the slope of the forgetting factor is also studied with the expectation that an increase in the forgetting factor slope l will make the Algorithm 5.2 forget slower. The experiments are re-run for $l = .6$ and .9 to determine the role of the forgetting factor and mean energy gains for Algorithm 5.2 over the LEACH algorithm are tabulated in Table 5.2. The standard deviation of the gain is also illustrated. Once again, the deviation is equal to approximately 1% and is small relative to the mean energy gain. This confirms that as l is increased from $.6 \to .8 \to .9$ the average gain is decreased. The mobile devices forget slower, and energy saving gains are reduced. The overall trend is that increasing p increases the energy saving gains for $p = .15$, but energy saving gains then decrease as p increases further. The fact that the best gains are achieved for a probability $p = .15$ and $l = .6$ motivates the need for a more in-depth study to determine the best p, l pairs for different deployments.

Another interesting observation is how the number of cluster-heads changes over time. A secondary study on the effect of p, l on the number of cluster-heads would help refine the deployment of the algorithms in different scenarios where different numbers of cluster-heads were preferable. This has been included as part of the future work. The ability of these algorithms to select different numbers of mobile devices to be cluster-heads is advantageous, because irrespective of the number of cluster-heads, X, significant energy savings are achieved.

5.5 Conclusion

The aim of the fifth research question (RQ5) stated in 1.1.5 was to investigate mechanisms for adapting the collaborative sensing framework for scenarios with limited communication technologies. This chapter has clearly shown how a clustering approach can easily be deployed in such a situation with overall energy savings.

The novel cluster-head trajectory minimisation procedure is replicated by both Algorithm CH-Trajectory (Algorithm 5.1) and Algorithm CH-Trajectory-With-Forgetting-Factor (Algorithm 5.2) to stochastically select the cluster-head by including location information. Additionally, the process of greedily optimizing the location of the next cluster-head by modulating individual mobile device thresholds, in conjunction with existing well-placed cluster-heads, causes the responsibility of cluster-headship to be transferred from well-placed to potentially better-placed mobile devices successively, with high probability. Both the algorithms can be easily be integrated into the collaborative framework with mobile devices in proximity communicating using WiFi-Direct, which allows peer-2-peer transmissions between the mobile devices and only some selected devices communicate the aggregated sensed data via cellular technology. The algorithms also have the added benefit of being relatively low latency, low bandwidth and energy-efficient.

Thus, the main contributions of this chapter are in the form of the two algorithms, Algorithm CH-Trajectory and Algorithm CH-Trajectory-With-Forgetting-Factor that present collaborative techniques in communication-limited scenarios whilst reducing the energy-consumption, when sensed information is offloaded to an application cloud server.

Chapter 6

Conclusion and Future Work

The increasingly sophisticated sensors embedded into mobile devices are harnessed by multiple applications for user personalization and context-awareness, in sectors such as healthcare, social networks, traffic managements and environmental monitoring. This growing demand for smartphones and tablets, introduces a number of challenges as applications deal with a higher degree of resource heterogeneity, intermittent and highly variable resource connectivity and availability. By exploiting Mobile Cloud Computing techniques, the deployment of such computation-intensive mobile applications can be accelerated with the powerful mobile cloud servers utilized for offloading storage and data processing operations. This further offers the benefits of conserving mobile handset resources including energy whilst meeting application performance targets.

The main objective of this dissertation was to develop and examine a collaborative mobile sensing framework for the aforementioned scenario to utilize the scalability and processing capabilities of the mobile cloud and provide efficient collection of the sensed data to multiple applications in term of the energy efficiency and monitoring accuracy. This has been formalized in the research hypothesis with five research questions presented in Chapter 1. The previous chapters of this dissertation have addressed each of these questions in detail providing insight into how the collaborative sensing framework compiles the various distinct but complimentary components.

This chapter focuses on concluding the foundational research work that developed solutions and techniques in the core areas of Mobile Sensing and Mobile Cloud Computing. It provides a short summary of the chapters followed by presenting future work.

6.1 Summary

Chapter 1 of the dissertation introduced the main core research fields of Mobile Sensing and Mobile Cloud computing. It outlined the motivation of this dissertation along with the research hypothesis and research questions. The next chapter, Chapter 2 was an in-depth study on the technology, trends and challenges of these fields along with mobility models for simulating movement. It also presented research in the relevant domains of collaboration amongst devices, aggregation, context-awareness, and clustering solutions which form important aspects of the work. Additionally, it presented the state-of-the-art algorithms that have been deployed in real world scenario.

Chapter 3 focused on the design of the collaborative sensing framework as a centralized middleware to optimally select embedded sensors from mobile devices for accurately satisfying application constraints in an energy-efficient manner. The framework acted as a mediation between the multiple applications, mobile devices and heterogeneous sensors either embedded in the device or surrounding the mobile user. A key challenge here was to balance the trade-off between the accuracy of the information received by the application logic with the volume of data offloaded (at significant energy cost) by the mobile devices. This was made possible with the Algorithm Info-Aggregation that eliminated redundancy in the offloaded data and used frequent pattern mining for aggregation to reduce the number of active mobile devices, so that data served by a given device can be served to multiple applications. Its performance was compared with Algorithm No-Aggregation that did not recognize the potential to use the sensed data to serve more than one applications. Two sets of results were consolidated, where the first set made the simplifying assumption that mobile devices are stationary. Consequently, the latter used the Truncated Levy walk Mobility Model to emulate mobile device movements. Final results showed how the Algorithm Info-Aggregation reduced the expended cumulative energy and the number of sensors activated in devices and used a lower number of transmitted messages by reducing volume of offloaded sensed data.

The next chapter, Chapter 4 presented the concept of using application specific state machines. These machines were used to encode the contexts required by the event-driven mobile applications and efficiently help in reducing energy expended during transmission of sensed data. The first section of this chapter focused on the enterprise environment where mobile handsets provided by an organisation to its employees are used to provide context information to centralised (enterprise cloud hosted) applications that monitor and control the work environment. The use of continuous sensing in such scenarios has been found to jeopardize the battery backup in mobile devices. The Algorithm Context-Localize succeeded in reducing the energy expended during sensing and uses a clustering hierarchy for quickly localizing significant context changes. The second section presented a more IoT driven scenario inspired by the continuous proliferation of new low-energy sensors, wearable technology and ubiquitous devices. Mobile devices were harnessed as travelling gateways to access the static IoT sensors surrounding the mobile user and the framework was extended to create consolidated state-machines depending on the requirements of multiple applications for small physical sensing areas. Algorithm Assisted-Aggregation used the frequent pattern mining approach combined with the application-specific state machines to quantify reporting time-points for the applications, thereby reducing the volume of sensed data offloaded along with the number of active devices in an energy-efficient manner.

The last chapter, Chapter 5 considered situations in which mobile devices expend additional energy when transmitting sensed data over cellular networks, due to unavailability of other communication mediums. In this scenario, clustering algorithms were adapted for collaboration between mobile devices in close proximity and added to the collaborative sensing framework. The approach was two cluster-head selection algorithms that optimized the transmission costs measured in terms of the communication distance between the mobiles and the cellular base station and the inter-mobile communication distances. These algorithms were designed to use the knowledge regarding the communication distance to modulate the probability of each mobile becoming a cluster-head. The algorithms were compared with LEACH, a cluster-based technique widely used for wireless sensor networks and saved residual energy of the devices without a penalty in terms of sensing accuracy.

6.2 Future Work

To conclude this dissertation, a discussion on the possible avenues for further research is presented. These offer the potential to improve performance of the presented algorithms in an energy-efficient manner.

A natural extension to the framework would be to provide scalable solutions for larger deployments of crowd sourcing applications, as also mentioned in §3.2. By using processed sensed data from the cloud, small deployments can be designed to function as local clusters of mobile devices, that use the framework as gateways to communicate and aggregate data. This exploits the potential of a hierarchical distributed scheme providing different levels of aggregation to the applications.

Additionally, partial covering and no-covering of requested locations by mobile devices can be included for the application as indicated in §3.3 and §4.3.1, by formalizing multiple options. This will ensure that different applications with different levels of data quality can be satisfied since no solution will be universally-applicable. One such option would be to select a set of closest devices outside the location boundary to send sensed data collectively. This would includes loss in accuracy depending on the distance of the mobile devices from the location. Another option would be to assume no data or missing data for the location. Mechanisms for improved calculation of the optimal selection of the offload percentage of data directly into the cloud, in comparison to the simple approach adapted in this dissertation §3.4, is also required.

Furthermore, enhanced data analysis and model improvement is required to completely understand the decrease that is noticed with cumulative energy gain and the trends observed with the number of active devices and volume of offloaded data, when Algorithm No-Aggregation is compared with Algorithm Info-Aggregation in §3.5 and when Algorithm Info-Aggregation is compared with Algorithm Assisted-Aggregation in §4.3.3. This analysis will allow an insight into the optimal number of mobile devices and embedded sensors present in the physical sensing area as application requests increase.

Although dynamic alteration of sensing rates has been attempted for one application in this work, the framework can be further extended such that sensing rates are also embedded into the application-specific state-machines for further energy improvements. The framework can also be adapted to incorporate false negative identification, more complex state machine diagrams and machine learning strategies to analyse application requirements for autonomous creation and improvement of state machines, as also mentioned in §4.3.2.

Advanced clustering approaches, combined with machine learning to further improve cluster-head selection for a generic scenario, will also be studied as part of the future work. This includes modelling and simulation of multiple IoT inspired scenarios, for example supporting efficient low-latency data collection from autonomous cars with vehicle-to-vehicle and vehicle-to-infrastructure communication. Multiple features can also be incorporated into the cluster-head trajectories for better energy-efficiency and fairness, along with in-depth analysis to obtain the best combination of the probability of a device to become the cluster-head, the slope of the forgetting function and the number of cluster-heads being selected, as also indicated in §5.4.

An important aspect of the future work would be the implementation of strategies to maintain trust and privacy amongst the mobile devices and the sensing framework. This includes identification of mobile devices in the network with malicious intent or those providing inaccurate/false sensed data, maintenance of precision of sensed data delivered to the applications, restriction on a device to join the network when posing a threat and ensuring encryption techniques so that sensitive data from devices cannot be leaked to other devices. Lastly, another extension of the framework is possible with the integration of a network simulator or a network interface to study issues relating to delay in messages, congestion in network and unavailable network on mobile devices due to their mobility pattern. Several network parameters can then be varied, including the bandwidth, to obtain traffic and congestion statistics. This would also include real-data analysis of mobile devices with different operating systems and capabilities to fulfil contrasting application constraints.

References

Saeid Abolfazli, Z Sanaei, M Sanaei, M Shojafar, and A Gani. Mobile cloud computing: The-state-of-the-art, challenges, and future research. *Encyclopedia of Cloud Computing, Wiley, USA*, 2015. 17

Murtadha Aldeer, Richard Howard, and Ahmed Al-Hilli. Minimizing energy consumption in transmit-only sensor networks via optimal placement of the cluster heads. In *Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop*, pages 36–38. ACM, 2016. 28

Mazhar Ali, Samee U Khan, and Athanasios V Vasilakos. Security in cloud computing: Opportunities and challenges. *Information Sciences*, 305:357–383, 2015. 21

Mojtaba Alizadeh, Saeid Abolfazli, Mazdak Zamani, Sabariah Baharun, and Kouichi Sakurai. Authentication in mobile cloud computing: A survey. *Journal of Network and Computer Applications*, 61:59–80, 2016. 21

Michele Amoretti, Luca Consolini, Alessandro Grazioli, and Francesco Zanichelli. Impact of different auto-scaling strategies on adaptive Mobile Cloud Computing systems. In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 589–596. IEEE, 2016. 18, 20

Ganesh Ananthanarayanan, Venkata N. Padmanabhan, Lenin Ravindranath, and Chandramohan A. Thekkath. COMBINE: Leveraging the Power of Wireless Peers Through Collaborative Downloading. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 286–298, New York, NY, USA, 2007. ACM. doi: 10.1145/1247660.1247693. 24, 27

Android Developers. CellLocation. URL http://developer.android.com/reference/android/telephony/CellLocation.html. Online: Last Accessed in 2015. 89

Pelin Angin, Bharat Bhargava, and Sumi Helal. A Mobile-Cloud Collaborative Traffic Lights Detector for Blind Navigation. In *2010 Eleventh International Conference on Mobile Data Management (MDM)*, pages 396 –401, May 2010. 19

Pelin Angin, Bharat Bhargava, and Zhongjun Jin. A Self-Cloning Agents Based Model for High-Performance Mobile-Cloud Computing. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 301–308. IEEE, 2015. 18

A. Anjum and M.U. Ilyas. Activity recognition using smartphone sensors. In *2013 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 914–919, January 2013. 26

Apollo. Discovery CO/Heat Multisensor Detector. URL http://www.apollo-fire.co.uk. Online: Last Accessed in 2014. 64

Inc. Apple. Apple-iphones. URL https://www.apple.com/iphone/. Online: Last Accessed in 2016. 51, 80, 104

N. Aschenbruck, E. Gerhards-Padilla, and P. Martini. A survey on mobility models for performance analysis in tactical mobile networks. *Journal of Telecommunications and Information Technology*, nr 2:54–61, 2008. 22

B. Atakan and O.B. Akan. Immune System Based Distributed Node and Rate Selection in Wireless Sensor Networks. In *Proc. 1st Int. Conf. Bio-Inspired Models of Network, Information and Computing Systems*, pages 1–8, December 2006. doi: 10.1109/BIMNICS.2006.361806. 28

Seung Jun Baek, G. de Veciana, and X. Su. Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation. *IEEE Journal on Selected Areas in Communications*, 22(6):1130–1140, August 2004. doi: 10.1109/JSAC.2004.830934. 25

Fan Bai and Ahmed Helmy. Chapter1: A survey of mobility models. *Wireless Adhoc Networks*, 206:147, 2004. 22

Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proceddings of the 9th ACM SIGCOMM Conference on Internet Measurements Confernce*, IMC, pages 280–293, USA, 2009. doi: 10.1145/1644893.1644927. 51, 81, 104

Payal Khurana Batra and Krishna Kant. LEACH-MAC: a new cluster head selection algorithm for Wireless Sensor Networks. *Wireless Networks*, 22(1):49–60, 2016. 28

Fehmi Ben Abdesslem, Andrew Phillips, and Tristan Henderson. Less is More: Energy-efficient Mobile Sensing with Senseless. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds*, MobiHeld '09, pages 61–62. ACM, 2009. doi: 10.1145/1592606.1592621. URL http://doi.acm.org/10.1145/1592606.1592621. 15, 27

V Bharath. Energy-Efficient Data Traffic Management in Mobile Cloud Computing. *Energy (Oxford)*, 2(4), 2014. ISSN 0360-5442. 27

Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the Internet of Things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, MCC '12, page 13–16, New York, NY, USA, 2012. ACM. URL http://doi.acm.org/10.1145/2342509.2342513. 17

Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006. 23

Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483—-502, 2002. ISSN 1530-8677. doi: 10.1002/wcm.72. 22

Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, and Ronald A. Peterson. People-centric urban sensing. In *Proceedings of the 2nd Annual International Workshop on Wireless Internet*, WICON, New York, NY, USA, 2006. ACM. doi: 10.1145/1234161.1234179. 13, 26

A.T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, R.A. Peterson, Hong Lu, Xiao Zheng, M. Musolesi, K. Fodor, and Gahng-Seop Ahn. The Rise of People-Centric Sensing. *IEE Internet Computing*, 12(4):12–21, July 2008. doi: 10.1109/MIC.2008.90. 13, 26

Valeria Cardellini, Vittoria De Nitto Personé, Valerio Di Valerio, Francisco Facchinei, Vincenzo Grassi, Francesco Lo Presti, and Veronica Piccialli. A game-theoretic approach to computation offloading in mobile cloud computing. *Mathematical Programming*, 157 (2):421–449, 2016. 18

Giuseppe Cardone, Antonio Corradi, Luca Foschini, and Raffaele Ianniello. ParticipAct: A Large-Scale Crowdsensing Platform. *IEEE Transactions on Emerging Topics in Computing*, 4(1):21–32, 2016. 16

Dimitris Chatzopoulos, Mahdieh Ahmadi, Sokol Kosta, and Pan Hui. Have you asked your neighbors? a hidden market approach for device-to-device offloading. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*, pages 1–9. IEEE, 2016. 21, 24

Min Chen, Yin Zhang, Yong Li, Shiwen Mao, and Victor CM Leung. EMC: emotion-aware mobile cloud computing in 5G. *IEEE Network*, 29(2):32–38, 2015. 26

Yuanzhu Chen, AL. Liestman, and Jiangchuan Liu. A hierarchical energy-efficient framework for data aggregation in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 55(3):789–796, May 2006. doi: 10.1109/TVT.2006.873841. 25

S. Chetan, Gautam Kumar, K. Dinesh, K. Mathew, and M. Abhimanyu. Cloud computing for Mobile World, 2010. URL http://chetan.ueuo.com/projects/CCMW.pdf. 20

Jason H. Christensen. Using RESTful web-services and cloud computing to create next generation mobile applications. In *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, OOPSLA ’09, pages 627—-634, New York, NY, USA, 2009. ACM. 17

Byung Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. CloneCloud: elastic execution between mobile device and cloud. In *Proceedings of the 6th Conference on Computer Systems*, EuroSys, pages 301—-314, New York, NY, USA, 2011. ACM. doi: 10.1145/1966445.1966473. 18

Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020 white paper. URL http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. Last Accessed in 2016. 1

Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: Privacy-aware People-centric Sensing. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys ’08, pages 211–224, New York, NY, USA, 2008. ACM. doi: 10.1145/1378600.1378624. 24

Preston A. Cox. Mobile cloud computing, 2011. URL http://www.ibm.com/developerworks/cloud/library/cl-mobilecloudcomputing/index.html. 21

Eduardo Cuervo, Aruna Balasubramanian, Dae Ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. MAUI: making smartphones last longer with code offload. In *Proceedings of the 8th International conference on Mobile Systems, Applications, and Services*, MobiSys, pages 49—-62, New York, USA, 2010. ACM. doi: 10.1145/1814433.1814441. 18

A. da Silva Rego, J. Celestino, A. dos Santos, E.C. Cerqueira, A. Patel, and M. Taghavi. BEE-C: A bio-inspired energy efficient cluster-based algorithm for data continuous dissemination in Wireless Sensor Networks. In *Proc. 18th IEEE Int. Conf. on Netw., ICON*, pages 405–410, December 2012. doi: 10.1109/ICON.2012.6506592. 28

Eraldo Damosso and COST Telecom Secretariat. Digital Mobile Radio Towards Future Generation Systems: COST 231 Final Report., 1999. URL http://http://www.lx.it.pt/cost231/final_report.htm. 104

K. Dasgupta, K. Kalpakis, and P. Namjoshi. An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1948–1953, March 2003. doi: 10.1109/WCNC.2003.1200685. 25

Shantanu Deshmukh and Rinku Shah. Computation offloading frameworks in mobile cloud computing: a survey. In *Current Trends in Advanced Computing (ICCTAC), IEEE International Conference on*, pages 1–5. IEEE, 2016. 18

S. Devadas and A. R. Newton. Decomposition and factorization of sequential finite state machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(11):1206–1217, November 1989. ISSN 0278-0070. doi: 10.1109/43.41505. 78

Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 2011. doi: 10.1002/wcm.1203. xv, 16, 17, 19, 20, 21

Falko Dressler and Ozgur B. Akan. A Survey on Bio-Inspired Networking. *Comp. Netw.*, 54 (6):881–900, April 2010. ISSN 1389-1286. doi: 10.1016/j.comnet.2009.10.024. 28

Akshay Dua, Nirupama Bulusu, Wu Chang Feng, and Wen Hu. Towards trustworthy participatory sensing. In *HotSec'09: Proceedings of the Usenix Workshop on Hot Topics in Security*, 2009. 13

Sourav Dutta, Ankur Narang, Souvik Bhattacherjee, Ananda Swarup Das, and Dilip Krishnaswamy. Predictive Caching Framework for Mobile Wireless Networks. In *2015 16th IEEE International Conference on Mobile Data Management*, volume 1, pages 179–184. IEEE, 2015. 26

Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. BikeNet: A Mobile Sensing System for Cyclist Experience Mapping. *ACM Transactions on Sensor Networks*, 6(1):6:1—6:39, January 2010. doi: 10.1145/1653760.1653766. 26

Hassan El Alami and Abdellah Najid. Energy-efficient fuzzy logic cluster head selection in wireless sensor networks. In *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, pages 1–7. IEEE, 2016. 28

ENAikoon. OpenCellID. URL http://opencellid.org/. Online: Last Accessed in 2015. 89

Heungsik Eom, Renato Figueiredo, Huaqian Cai, Ying Zhang, and Gang Huang. MALMOS: Machine Learning-Based Mobile Offloading Scheduler with Online Training. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*, pages 51–60. IEEE, 2015. 18

Jingyao Fan, Qinghua Li, and Guohong Cao. Privacy-aware and trustworthy data aggregation in mobile sensing. In *Communications and Network Security (CNS), 2015 IEEE Conference on*, pages 31–39. IEEE, 2015. 25

Yujie Fan, Yanfang Ye, and Lifei Chen. Malicious sequential pattern mining for automatic malware detection. *Expert Systems with Applications*, 52:16–25, 2016. 26

Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos. TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1231–1239. IEEE, 2014. 24

D. Fotue, F. Melakessou, T. Engel, and H. Labiod. Design of New Aggregation Techniques for Wireless Sensor Networks. In *Proceedings of the IEEE International Symposium on*

Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), pages 400–402, 2010. doi: 10.1109/MASCOTS.2010.70. 25

R. Friedman, A. Kogan, and Y. Krivolapov. On Power and Throughput Tradeoffs of WiFi and Bluetooth in Smartphones. *IEEE Transactions on Mobile Computing*, 12(7): 1363–1376, July 2013. ISSN 1536-1233. doi: 10.1109/TMC.2012.117. 51, 81

Keke Gai, Meikang Qiu, Hui Zhao, Lixin Tao, and Ziliang Zong. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, 59:46–54, 2016. 20

L. Galluccio, S. Palazzo, and A.T. Campbell. Efficient data aggregation in wireless sensor networks: An entropy-driven analysis. In *Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC)*, pages 1–6, 2008. doi: 10.1109/PIMRC.2008.4699915. 25

Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011. 27

Lin Gao, Fen Hou, and Jianwei Huang. Providing long-term participation incentive in participatory sensing. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2803–2811. IEEE, 2015. 24

Louis Giglio, Jacques Descloitres, Christopher O. Justice, and Yoram J. Kaufman. An Enhanced Contextual Fire Detection Algorithm for MODIS. *Remote Sensing of Environment*, 87(2–3):273–282, October 2003. 27

Marta C. González, César A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008. doi: 10.1038/nature06958. 23

Google. Google street view. URL https://www.google.com/maps/views/streetview?hl=en&gl=ie. Online: Last Accessed in 2016. 14

Le Guan, Xu Ke, Meina Song, and Junde Song. A Survey of Research on Mobile Cloud Computing. In *2011 IEEE/ACIS 10th International Conference on Computer and Information Science (ICIS)*, pages 387 –392, May 2011. 20, 21

Yang Guan, Yao Xiao, Leonard J Cimini, and Chien-Chung Shen. Power efficient peer-to-peer streaming to co-located mobile users. In *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pages 321–326. IEEE, 2014. 20

Bin Guo, Huihui Chen, Zhiwen Yu, Xing Xie, Shenlong Huangfu, and Daqing Zhang. Fliermeet: a mobile crowdsensing system for cross-space public information reposting, tagging, and sharing. *IEEE Transactions on Mobile Computing*, 14(10):2020–2033, 2015. 14, 16

Saurav Gupta, Sanjay P Sood, and DK Jain. 'Let's Exercise': A Context Aware Mobile Agent for Motivating Physical Activity. In *Emerging Research in Computing, Information, Communication and Applications*, pages 511–520. Springer, 2016. 26

Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining Frequent Patterns Without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53—87, 2004. 41, 45, 78

Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, April 2006. 25, 41, 45, 78

Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55—-86, August 2007a. doi: 10.1007/s10618-006-0059-1. 25, 41, 78

Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.*, 15(1):55—-86, August 2007b. 25

Qilong Han, Shuang Liang, and Hongli Zhang. Mobile cloud sensing, big data, and 5g networks make an intelligent and smart world. *IEEE Network*, 29(2):40–45, 2015. 21

M.J. Handy, M. Haase, and D. Timmermann. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In *Proc. 4th Int. Wk. Shp. on Mobi. and Wireless Comm. Net.*, pages 368–72, 2002. doi: 10.1109/MWCN.2002.1045790. 27

W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Comm.*, 1(4):660–70, 2002. doi: 10.1109/TWC.2002.804190. 28

W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. 33rd Annu. Hawaii Int. Conf. on Sys. Sci.*, 2000. doi: 10.1109/HICSS.2000.926982. 27, 98

D.B. Hoang and Lingfeng Chen. Mobile Cloud for Assistive Healthcare (MoCAsH). In *Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific*, pages 325 –332, December 2010. 19

Steven A. Hofmeyr and Stephanie Forrest. Architecture for an Artificial Immune System. *Evolutionary Computation*, 8(4):443–473, December 2000. ISSN 1063-6560. doi: 10.1162/106365600568257. 28

Intel. Urban Atmospheres. URL http://www.urban-atmospheres.net/. Online: Last Accessed in 2016. 14

Hiroshi Ishida, Yukihiko Kagawa, Takamichi Nakamoto, and Toyosaka Moriizumi. Odor-source localization in the clean room by an autonomous mobile sensing system. *Sensors and Actuators B: Chemical*, 33(1–3):115–121, July 1996. 26

Hiroshi Ishida, Takamichi Nakamoto, and Toyosaka Moriizumi. Remote sensing of gas/odor source location and concentration distribution using mobile system. *Sensors and Actuators B: Chemical*, 49(1–2):52–57, June 1998. 27

VS Jagtap, Karishma Vikas Pawar, and Ajeet Ram Pathak. Augmented Execution in Mobile Cloud Computing: A Survey. In *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*, pages 237–244. IEEE, 2014. 18

Yaser Jararweh, Fadi Ababneh, Abdallah Khreishah, Fahd Dosari, et al. Scalable cloudlet-based mobile computing model. *Procedia Computer Science*, 34:434–441, 2014. 20

Paulo Jesus, Carlos Baquero, and Paulo Sérgio Almeida. A survey of distributed data aggregation algorithms. *IEEE Communications Surveys & Tutorials*, 17(1):381–404, 2015. 25

An Jian, Gui Xiaolin, Yang Jianwei, Sun Yu, and He Xin. Mobile crowd sensing for internet of things: A credible crowdsourcing model in mobile-sense service. In *Multimedia Big Data (BigMM), 2015 IEEE International Conference on*, pages 92–99. IEEE, 2015. xv, 14, 27

Haiming Jin, Lu Su, Houping Xiao, and Klara Nahrstedt. Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM*, page 341, 2016. 25

Xin Jin and Yu-Kwong Kwok. Cloud Assisted P2P Media Streaming for Bandwidth Constrained Mobile Subscribers. In *2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 800 –805, December 2010. 20

Eric Jung, Yichuan Wang, Iuri Prilepov, Frank Maker, Xin Liu, and Venkatesh Akella. User-profile-driven collaborative bandwidth sharing on mobile phones. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, MCS '10, New York, NY, USA, 2010. ACM. 20

Hoill Jung, Kyung-Yong Chung, and Young-Ho Lee. Decision supporting method for chronic disease patients based on mining frequent pattern tree. *Multimedia Tools and Applications*, 74(20):8979–8991, 2015. 26

Rakpong Kaewpuang, Dusit Niyato, Ping Wang, and Ekram Hossain. A framework for cooperative resource management in mobile cloud computing. *IEEE Journal on Selected Areas in Communications*, 31(12):2685–2700, 2013. 24

Seungwoo Kang, Jinwon Lee, Hyukjae Jang, Hyonik Lee, Youngki Lee, Souneil Park, Taiwoo Park, and Junehwa Song. SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 267–280, New York, NY, USA, 2008. ACM. doi: 10.1145/1378600.1378630. 15, 27

S.H. Kang and Thinh Nguyen. Distance Based Thresholds for Cluster Head Selection in Wireless Sensor Networks. *IEEE Comm. Lett.*, 16(9):1396–9, 2012. doi: 10.1109/LCOMM.2012.073112.120450. 28

D. Karamshuk, Chiara Boldrini, M. Conti, and A. Passarella. Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165, December 2011. doi: 10.1109/MCOM.2011.6094021. 42

Mahir Kaya, Altan Koçyiğit, and P Erhan Eren. An adaptive mobile cloud computing framework using a call graph based model. *Journal of Network and Computer Applications*, 65:12–35, 2016. 18

W.Z. Khan, Yang Xiang, M.Y. Aalsalem, and Q. Arshad. Mobile Phone Sensing Systems: A Survey. *IEEE Communications Surveys Tutorials*, 15(1):402–427, 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.031412.00077. 2, 13, 14, 26, 68

Seung-Woo Ko, Kaibin Huang, Seong-Lyun Kim, and Hyukjin Chae. Online Prefetching for Mobile Computation Offloading. *arXiv preprint arXiv:1608.04878*, 2016. 20

Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proceedings of IEEE INFOCOM*, pages 945–953. IEEE, 2012. doi: 10.1109/INFCOM.2012.6195845. 18

Ioannis Krontiris and Tassos Dimitriou. A platform for privacy protection of data requesters and data providers in mobile sensing. *Computer Communications*, 65:43–54, 2015. 15

K. Kumar and Yung Hsiang Lu. Cloud Computing for Mobile Users: Can Offloading Computation Save Energy? *Computer*, 43(4):51–56, April 2010. doi: 10.1109/MC.2010.98. 2, 18

Pradeep Kumar, P. Radha Krishna, and S. Bapi Raju. *Pattern Discovery Using Sequence Data Mining: Applications and Studies*. Idea Group Inc (IGI), September 2011. 25

Marc Kurz, Gerold Holzl, Alois Ferscha, Alberto Calatroni, Daniel Roggen, Gerhard Troster, Hesam Sagha, Ricardo Chavarriaga, Jose del R. Millan, David Bannach, Kai Kunze, and Paul Lukowicz. The Opportunity Framework and Data Processing Ecosystem for Opportunistic Activity and Context Recognition. *International Journal of Sensors Wireless Communications and Control*, 1(2):102–125, December 2011. 26, 59

Nicholas D Lane and Petko Georgiev. Can deep learning revolutionize mobile sensing? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 117–122. ACM, 2015. 15

Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T. Campbell. Urban sensing systems: opportunistic or participatory? In *Proceedings of the 9th workshop on Mobile computing systems and applications*, HotMobile '08, pages 11—-16, New York, NY, USA, 2008. ACM. doi: 10.1145/1411759.1411763. 1, 13, 14

Nicholas D. Lane, E. Miluzzo, Hong Lu, D. Peebles, T. Choudhury, and A.T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150, 2010. doi: 10.1109/MCOM.2010.5560598. 2, 12, 26, 70

Nicholas D Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. An Early Resource Characterization of Deep Learning on Wearables, Smartphones and Internet-of-Things Devices. In *Proceedings of the 2015 International Workshop on Internet of Things towards Applications*, pages 7–12. ACM, 2015. 15

N.M.A. Latiff, C.C. Tsimenidis, and B.S. Sharif. Energy-Aware Clustering for Wireless Sensor Networks using Particle Swarm Optimization. In *Proc. 18th IEEE Int. Symp. on Personal, Indoor and Mob. Radio Comm., PIMRC*, pages 1–5, September 2007. doi: 10.1109/PIMRC.2007.4394521. 28

Steven Latré, Philip Leroux, Tanguy Coenen, Bart Braem, Pieter Ballon, and Piet Demeester. City of things: An integrated and multi-technology testbed for iot smart city experiments. In *Smart Cities Conference (ISC2), 2016 IEEE International*, pages 1–8. IEEE, 2016. 68

Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 151–165. ACM, 2015. 20

Lei Lei, Zhangdui Zhong, Kan Zheng, Jiadi Chen, and Hanlin Meng. Challenges on wireless heterogeneous networks for mobile cloud computing. *IEEE Wireless Communications*, 20 (3):34–44, 2013. 21

Jiuyong Li, Ada Wai-chee Fu, Hongxing He, Jie Chen, Huidong Jin, Damien McAullay, Graham Williams, Ross Sparks, and Chris Kelman. Mining risk patterns in medical data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 770—775, New York, NY, USA, 2005. ACM. 26

Qinghua Li and Guohong Cao. Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 60–81. Springer, 2013. 25

Qinghua Li and Guohong Cao. Providing Privacy-Aware Incentives in Mobile Sensing Systems. *IEEE Transactions on Mobile Computing*, 15(6):1485–1498, 2016. 15

Qinghua Li, Guohong Cao, and Thomas F La Porta. Efficient and privacy-aware data aggregation in mobile sensing. *IEEE Transactions on dependable and secure computing*, 11(2):115–129, 2014. 25

Wenzhong Li, Yanchao Zhao, Sanglu Lu, and Daoxu Chen. Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing. *IEEE Communications Magazine*, 53(3):89–97, 2015a. 21

Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. On the discovery of evolving truth. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 675–684. ACM, 2015b. 16

Daniyal Liaqat, Silviu Jingoi, Eyal de Lara, Ashvin Goel, Wilson To, Kevin Lee, Italo De Moraes Garcia, and Manuel Saldana. Sidewinder: An Energy Efficient and Developer Friendly Heterogeneous Architecture for Continuous Mobile Sensing. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 205–215. ACM, 2016. 27

Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao. Energy-accuracy Trade-off for Continuous Mobile Device Location. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 285–298, New York, NY, USA, 2010. ACM. doi: 10.1145/1814433.1814462. 15, 23, 27

Cihang Liu, Lan Zhang, Zongqian Liu, Kebin Liu, Xiangyang Li, and Yunhao Liu. Lasagna: towards deep hierarchical understanding and searching over mobile sensing data. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 334–347. ACM, 2016. 15

Fangming Liu, Peng Shu, and John CS Lui. Appatp: An energy conserving adaptive mobile-cloud transmission protocol. *IEEE Transactions on Computers*, 64(11): 3051–3063, 2015a. 21, 24

Jieyao Liu, Ejaz Ahmed, Muhammad Shiraz, Abdullah Gani, Rajkumar Buyya, and Ahsan Qureshi. Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions. *Journal of Network and Computer Applications*, 48:99–117, 2015b. 18

Radhika Loomba, Lei Shi, and Brendan Jennings. State-machine driven opportunistic sensing by mobile devices. In *IEEE Global Communications Conference (GLOBECOM)*, pages 2739–2744, December 2014a. doi: 10.1109/GLOCOM.2014.7037222. xxiii, 58

Radhika Loomba, Lei Shi, Brendan Jennings, Roy Friedman, John Kennedy, and Joe Butler. Information Aggregation for Collaborative Sensing in Mobile Cloud Computing. In *2nd*

IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pages 149–158, April 2014b. doi: 10.1109/MobileCloud.2014.23. xxiii, 32

Radhika Loomba, Ruairí de Fréin, and Brendan Jennings. Selecting Energy Efficient Cluster-Head Trajectories for Collaborative Mobile Sensing. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, December 2015a. xxiii, 88

Radhika Loomba, Lei Shi, Brendan Jennings, Roy Friedman, John Kennedy, and Joe Butler. Energy-aware collaborative sensing for multiple applications in mobile cloud computing. *Sustainable Computing: Informatics and Systems*, 8:47–59, December 2015b. ISSN 2210-5379. doi: 10.1016/j.suscom.2015.09.001. URL http://www.sciencedirect.com/science/article/pii/S2210537915000384. xxiii, 32

Radhika Loomba, Lei Shi, and Brendan Jennings. State-machine driven Collaborative Mobile Sensing Serving Multiple Internet-of-Things Applications. In *15th IFIP/IEEE International Symposium on Integrated Network Management (IM 2017) -Special Track on Management of IoT*, May 2017. to appear. xxiii, 58

Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, New York, NY, USA, 2010. ACM. doi: 10.1145/1869983.1869992. 15, 27

Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T. Chittaranjan, Andrew T. Campbell, Daniel Gatica-Perez, and Tanzeem Choudhury. StressSense: detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 351—-360, New York, NY, USA, 2012. ACM. doi: 10.1145/2370216.2370270. 14

Guoqiang Mao, Barış Fidan, and Brian D. O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, July 2007. 26

Martina Marjanović, Lea Skorin-Kapov, Krešimir Pripužić, Aleksandar Antonić, and Ivana Podnar Žarko. Energy-aware and quality-driven sensor management for green mobile crowd sensing. *Journal of Network and Computer Applications*, 59:95–108, 2016. 26

Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, and Kui Ren. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 183–196. ACM, 2015. 15

Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, and Andrew T. Campbell. CenceMe:Injecting Sensing Presence into Social Networking Applications. In *Proceedings of the 2Nd European Conference on Smart Sensing and Context*, EuroSSC'07, pages 1–28, Berlin, Heidelberg, 2007. Springer-Verlag. 14

Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 337–350, New York, USA, 2008. ACM. doi: 10.1145/1460412.1460445. 5, 15

Emiliano Miluzzo, Cory T Cornelius, Ashwin Ramaswamy, Tanzeem Choudhury, Zhigang Liu, and Andrew T Campbell. Darwin phones: the evolution of sensing and inference on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 5–20. ACM, 2010. 15

Sudip Misra, Snigdha Das, Manas Khatua, and Mohammad S Obaidat. Qos-guaranteed bandwidth shifting and redistribution in mobile cloud environment. *IEEE Transactions on Cloud Computing*, 2(2):181–193, 2014. 20

Karan Mitra, Arkady Zaslavsky, and Christer Åhlund. Context-aware QoE modelling, measurement, and prediction in mobile computing systems. *IEEE Transactions on Mobile Computing*, 14(5):920–936, 2015. 26

Abderrahmen Mtibaa, Afnan Fahim, Khaled A. Harras, and Mostafa H. Ammar. Towards Resource Sharing in Mobile Device Clouds: Power Balancing Across Mobile Devices. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing*, MCC, pages 51–56, USA, 2013. ISBN 978-1-4503-2180-8. doi: 10.1145/2491266.2491276. 21, 24

Abderrahmen Mtibaa, Khaled A Harras, Karim Habak, Mostafa Ammar, and Ellen W Zegura. Towards mobile opportunistic computing. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 1111–1114. IEEE, 2015. 21, 24

Min Mun, Shuai Hao, Nilesh Mishra, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Ramesh Govindan. Personal Data Vaults: A Locus of Control for Personal Data Streams. In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 17:1–17:12, New York, NY, USA, 2010. ACM. doi: 10.1145/1921168.1921191. 24

Nitin Naik and Paul Jenkins. A Secure Mobile Cloud Identity: Criteria for Effective Identity and Access Management Standards. In *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 89–90. IEEE, 2016. 21

S. Nath. ACE: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing. *IEEE Transactions on Mobile Computing*, 12(8):1472–1486, August 2013. doi: 10.1109/TMC.2013.12. 15, 27

T. Penner, A. Johnson, B. Van Slyke, M. Guirguis, and Qijun Gu. Transient clouds: Assignment and collaborative execution of tasks on mobile devices. In *Proceedings of the IEEE Global Communication Conference (GLOBECOM)*, pages 2801–2806, 2014. doi: 10.1109/GLOCOM.2014.7037232. 24

W.D. Phillips and R. Sankar. Improved transient weather reporting using people centric sensing. In *Proceedings of the 2013 IEEE Consumer Communications and Networking Conference (CCNC 2013)*, pages 920–925, January 2013. 27

B. Priyantha, D. Lymberopoulos, and Jie Liu. LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones. *IEEE Pervasive Computing*, 10(2):12–15, April 2011. doi: 10.1109/MPRV.2011.28. 15, 27, 51, 81

Aveek Purohit, Zheng Sun, Frank Mokaya, and Pei Zhang. Sensorfly: Controlled-mobile sensing platform for indoor emergency response applications. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 223–234. IEEE, 2011. 27

A Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy. Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications. In *Proceddings of the 77th IEEE Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2013. doi: 10.1109/VTCSpring.2013.6692723. 80, 89

A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy. 3GPP LTE traffic offloading onto WiFi Direct. In *Proceedings of the IEEE Wireless Communication and Networks*

Conference Workshops (WCNCW), pages 135–40, 2013. doi: 10.1109/WCNCW.2013.6533328. 80, 89

Kiran K. Rachuri, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow. SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 73–84, New York, NY, USA, 2011. ACM. doi: 10.1145/2030613.2030623. 15, 27

M. Reza Rahimi, Jian Ren, Chi Harold Liu, Athanasios V. Vasilakos, and Nalini Venkatasubramanian. Mobile Cloud Computing: A Survey, State of Art and Future Directions. *Mobile Networks and Applications*, 19(2):133–143, November 2014. doi: 10.1007/s11036-013-0477-4. 17

M. Rahmes, K. Wilder, K. Fox, and R. Pemble. A game theory model for situation awareness and management. In *Proceedings of the 2013 IEEE Consumer Communications and Networking Conference (CCNC 2013)*, pages 909–913, January 2013. 27

K. Ramesh and Dr K. Somasundaram. A comparative study of clusterhead selection algorithms in wireless sensor networks. *Int. J. of Comp. Sci. & Eng. Sur.*, 2(4):153–64, November 2011. doi: 10.5121/ijcses.2011.2411. 27

Rajib Rana, Chun Tung Chou, Nirupama Bulusu, Salil Kanhere, and Wen Hu. Ear-Phone: A context-aware noise mapping using smart phones. *Pervasive and Mobile Computing*, 17: 1–22, 2015. 26

S.S. Ranjani, S.R. Krishnan, and C. Thangaraj. Energy-efficient Cluster Based Data Aggregation for Wireless Sensor Networks. In *Proceedings of the International Conference on Recent Advances in Computing and Software Systems (RACSS)*, pages 174–179, 2012. doi: 10.1109/RACSS.2012.6212719. 25

M Mazhar Rathore, Awais Ahmad, Anand Paul, and Seungmin Rho. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks*, 101:63–80, 2016. 68

Kaushik Ravichandran, Ada Gavrilovska, and Santosh Pande. PiMiCo: Privacy Preservation via Migration in Collaborative Mobile Clouds. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 5341–5351. IEEE, 2015. 24

Abdul Razaque, Satwic Mudigulam, Kiran Gavini, Fathi Amsaad, Musbah Abdulgader, and Gondal Sai Krishna. H-LEACH: Hybrid-low energy adaptive clustering hierarchy for wireless sensor networks. In *Long Island Systems, Applications and Technology Conference (LISAT), 2016 IEEE*, pages 1–4. IEEE, 2016. 27

Francesco Restuccia, Sajal K Das, and Jamie Payton. Incentive mechanisms for participatory sensing: Survey and research challenges. *ACM Transactions on Sensor Networks (TOSN)*, 12(2):13, 2016. 13

Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the Levy-walk Nature of Human Mobility. *IEEE/ACM Transactions on Networking*, 19:630—-643, June 2011. doi: 10.1109/TNET.2011.2120618. 23, 42, 50, 80, 104

Atslands R. Rocha, Luci Pirmez, Flávia C. Delicato, írico Lemos, Igor Santos, Danielo G. Gomes, and José Neuman de Souza. WSNs Clustering Based on Semantic Neighborhood Relationships. *Computer Networks*, 56(5):1627–645, 2012. ISSN 1389-1286. doi: 10.1016/j.comnet.2012.01.014. 28

Alexey Rudenko, Peter Reiher, Gerald J. Popek, and Geoffrey H. Kuenning. Saving portable computer battery power through remote process execution. *SIGMOBILE Mobile Computing and Communications Review*, 2(1):19—-26, January 1998. doi: 10.1145/584007.584008. 2, 17

Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rajkumar Buyya. Heterogeneity in mobile cloud computing: taxonomy and open challenges. *IEEE Communications Surveys & Tutorials*, 16(1):369–392, 2014. 21

T.K. Sarkar, Zhong Ji, Kyungjung Kim, A. Medouri, and M. Salazar-Palma. A survey of various propagation models for mobile communication. *IEEE Ant. and Prop. Mag.*, 45(3): 51–82, 2003. ISSN 1045-9243. 104

S. Sarker, A. K. Nath, and A. Razzaque. Tradeoffs between sensing quality and energy efficiency for context monitoring applications. In *2016 International Conference on Networking Systems and Security (NSysS)*, pages 1–7, January 2016. doi: 10.1109/NSysS.2016.7400699. 15, 27

M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4):14 –23, Oct.-Dec. 2009. 2, 16, 20

Florian Schaub, Bastian Könings, and Michael Weber. Context-adaptive privacy: Leveraging context awareness to support privacy decision making. *IEEE Pervasive Computing*, 14(1): 34–43, 2015. 26

S. Selvakennedy, Sukunesan Sinnappan, and Yi Shang. T-ANT: A Nature-Inspired Data Gathering Protocol for Wireless Sensor Networks. *Jour. of Comm.*, 1(2), May 2006. ISSN 1796-2021. doi: 10.4304/jcm.1.2.22-29. 28

Senseable City Laboratory, MIT. Senseable City. URL http://senseable.mit.edu/. Online: Last Accessed in 2016. 14

Sensirion. New dimensions in environmental sensing, a. URL http://www.sensirion.com/en/mobile-solutions/environmental-sensing/. Online: Last Accessed in 2014. 51, 81

Sensirion. Digital humidity sensor sht2x (rh/t), b. URL https://www.sensirion.com/en/products/humidity-sensors/humidity-temperature-sensor-sht2x-digital-i2c-accurate/. Online: Last Accessed in 2016. 51, 81

Sensirion. Mobile solutions, c. URL https://www.sensirion.com/complete-sensor-solutions-for-environmental-sensing/. Online: Last Accessed in 2016. 12, 51, 81, 104

H. Shah-Mansouri and V.W.S. Wong. iCoMe: A novel incentivized cooperative mobile resource management mechanism. In *Proceedings of the IEEE Global Communication Conference (GLOBECOM)*, pages 4996–5001, December 2014. doi: 10.1109/GLOCOM.2014.7037597. 24

Ashish Sharma, Vishnu Navda, Ramachandran Ramjee, Venkat Padmanabhan, and Elizabeth Belding. Cool-Tether: Energy Efficient On-the-fly WiFi Hot-spots using Mobile Phones. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 109–120. Association for Computing Machinery, Inc., December 2009. 27

Rohini Sharma, Narendra Mishra, and Sumit Srivastava. A proposed energy efficient Distance Based Cluster Head (DBCH) algorithm: An improvement over LEACH. *Procedia Computer Science*, 57:807–814, 2015. 28

Cong Shi, Mostafa H. Ammar, Ellen W. Zegura, and Mayur Naik. Computing in cirrus clouds: the challenge of intermittent connectivity. In *Proceedings of the first edition of the*

MCC workshop on Mobile cloud computing, MCC '12, page 23–28, New York, NY, USA, 2012a. ACM. doi: 10.1145/2342509.2342515. 2, 16

Cong Shi, Vasileios Lakafosis, Mostafa H. Ammar, and Ellen W. Zegura. Serendipity: Enabling Remote Computing Among Intermittently Connected Mobile Devices. In *Proceedings of the 13th ACM International Symposium on Mobile Ad-Hoc Networking and Computing*, MobiHoc, pages 145–154, USA, 2012b. ISBN 978-1-4503-1281-3. doi: 10.1145/2248371.2248394. 21, 24

Minho Shin, Cory Cornelius, Dan Peebles, Apu Kapadia, David Kotz, and Nikos Triandopoulos. AnonySense: A System for Anonymous Opportunistic Sensing. *Pervasive and Mobile Computing*, 7(1):16–30, 2011. 24

Muhammad Shiraz, Mehdi Sookhak, Abdullah Gani, and Syed Adeel Ali Shah. A study on the critical analysis of computational offloading frameworks for mobile cloud computing. *Journal of Network and Computer Applications*, 47:47–60, 2015. 18

Peng Shu, Fangming Liu, Hai Jin, Min Chen, Feng Wen, Yupeng Qu, and Bo Li. etime: energy-efficient transmission between cloud and mobile devices. In *INFOCOM, 2013 Proceedings IEEE*, pages 195–199. IEEE, 2013. 24

Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, February 2010. doi: 10.1126/science.1177170. 23

Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009, October 2009. 25

Xiang Sun and Nirwan Ansari. Energy-optimized bandwidth allocation strategy for mobile cloud computing in LTE networks. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2120–2125. IEEE, 2015. 20

L. Tang, S. He, and Q. Li. Double-sided Bidding Mechanism for Resource Sharing in Mobile Cloud. *IEEE Transactions on Vehicular Technology*, PP(99), 2016. ISSN 0018-9545. 24

Mati B Terefe, Heezin Lee, Nojung Heo, Geoffrey C Fox, and Sangyoon Oh. Energy-efficient multisite offloading policy using markov decision process for mobile cloud computing. *Pervasive and Mobile Computing*, 27:75–89, 2016. 18

M.C.M. Thein and T. Thein. An Energy Efficient Cluster-Head Selection for Wireless Sensor Networks. In *Proc. Int. Conf. on Intel. Sys., Modelling and Sim. (ISMS)*, pages 287–91, 2010. doi: 10.1109/ISMS.2010.60. 28

J. Timmis, M. Neal, and J. Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1):143–150, 2000. 28

Wei Tu, Lei Wei, Wenyan Hu, Zhengguo Sheng, Hasen Nicanfar, Xiping Hu, Edith C-H Ngai, and Victor CM Leung. A Survey on Mobile Sensing Based Mood-Fatigue Detection for Drivers. In *Smart City 360°*, pages 3–15. Springer, 2016. 26

Moshe Unger. Latent Context-Aware Recommender Systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 383–386. ACM, 2015. 15

R Velmani and B Kaarthick. An efficient cluster-tree based data collection scheme for large mobile wireless sensor networks. *IEEE Sensors journal*, 15(4):2377–2390, 2015. 25

H. Wang, D. Peng, W. Wang, H. Sharif, J. Wegiel, D. Nguyen, R. Bowne, and C. Backhaus. Artificial Immune System based image pattern recognition in energy efficient Wireless Multimedia Sensor Networks. In *Proc. IEEE Military Comm. Conf., MILCOM*, pages 1–7, November 2008. doi: 10.1109/MILCOM.2008.4753651. 28

Xiaofei Wang and Min Chen. PreFeed: Cloud-Based Content Prefetching of Feed Subscriptions for Mobile Users. *IEEE Systems Journal*, 8(1):202–207, March 2014. doi: 10.1109/JSYST.2013.2265162. 27

Yating Wang, Ray Chen, and Ding-Chau Wang. A survey of mobile cloud computing applications: perspectives and challenges. *Wireless Personal Communications*, 80(4): 1607–1623, 2015. 19

Jiahui Wen, Mingyang Zhong, and Zhiying Wang. Activity recognition with weighted frequent patterns mining in smart environments. *Expert Systems with Applications*, 42(17): 6423–6432, 2015. 26

Dan Wu, Jinlong Wang, Rose Qingyang Hu, Yueming Cai, and Liang Zhou. Energy-efficient resource sharing for mobile device-to-device multimedia communications. *IEEE Transactions on Vehicular Technology*, 63(5):2093–2103, 2014. 24

Changj Xian, Yung-Hsiang Lu, and Zhiyuan Li. Adaptive computation offloading for energy conservation on battery-powered systems. In *2007 International Conference on Parallel and Distributed Systems*, volume 2, pages 1 –8, December 2007. 18

Liu Xiang, Jun Luo, and A Vasilakos. Compressed data aggregation for energy efficient wireless sensor networks. In *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 46–54, June 2011. doi: 10.1109/SAHCN.2011.5984932. 25

M. Xiao, J. Wu, H. Huang, L. Huang, and C. Hu. Deadline-Sensitive User Recruitment for Probabilistically Collaborative Mobile Crowdsensing. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 721–722, June 2016. doi: 10.1109/ICDCS.2016.15. 24

Mande Xie, Urmila Bhanja, Guoping Zhang, Guiyi Wei, and Yun Ling. Social role-based secure large data objects dissemination in mobile sensing environment. *Computer Communications*, 65:27–34, 2015a. 16

Zhenzhen Xie, Liang Hu, Feng Wang, Jin Li, and Kuo Zhao. PEMM: A Privacy-Aware Data Aggregation Solution for Mobile Sensing Networks. In *International Symposium on Intelligence Computation and Applications*, pages 474–482. Springer, 2015b. 25

Xi Xu, Rashid Ansari, Ashfaq Khokhar, and Athanasios V. Vasilakos. Hierarchical Data Aggregation Using Compressive Sensing (HDACS) in WSNs. *ACM Transactions on Sensor Networks*, 11(3):45:1–45:25, 2015. doi: 10.1145/2700264. 25

Jean Yang, Kuat Yessenov, and Armando Solar-Lezama. A Language for Automatically Enforcing Privacy Policies. *SIGPLAN Not.*, 47(1):85–96, January 2012. doi: 10.1145/2103621.2103669. 24

Haibo Ye, Tao Gu, Xiaorui Zhu, Jinwei Xu, Xianping Tao, Jian Lu, and Ning Jin. Ftrack: Infrastructure-free floor localization via mobile phone sensing. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 2–10. IEEE, 2012. 26

Chuang Wen You, Martha Montes-de Oca, Thomas J. Bao, Nicholas D. Lane, Hong Lu, Giuseppe Cardone, Lorenzo Torresani, and Andrew T. Campbell. CarSafe demo: supporting driver safety using dual-cameras on smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 547–547, New York, NY, USA, 2012. ACM. doi: 10.1145/2370216.2370304. 14

Özgür Yürür, Chi Harold Liu, Charith Perera, Min Chen, Xue Liu, and Wilfrido Moreno. Energy-efficient and context-aware smartphone sensor employment. *IEEE Transactions on Vehicular Technology*, 64(9):4230–4244, 2015. 26

Rui Zhang, Jing Shi, Yanchao Zhang, and Chi Zhang. Verifiable privacy-preserving aggregation in people-centric urban sensing systems. *IEEE Journal on Selected Areas in Communications*, 31(9):268–278, 2013. 25

Weishan Zhang, Shouchao Tan, Feng Xia, Xiufeng Chen, Zhongwei Li, Qinghua Lu, and Su Yang. A survey on decision making for task migration in mobile cloud environments. *Personal and Ubiquitous Computing*, 20(3):295–309, 2016a. 18

Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. Incentives for Mobile Crowd Sensing: A Survey. *IEEE Communications Surveys & Tutorials*, 18(1):54–67, 2016b. 12, 14

Dong Zhao, Huadong Ma, Shaojie Tang, and Xiang-Yang Li. Coupon: A cooperative framework for building sensing maps in mobile opportunistic networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(2):392–402, 2015. 16

Bowen Zhou, Amir Vahid Dastjerdi, Rodrigo N Calheiros, Satish Narayana Srirama, and Rajkumar Buyya. A context sensitive offloading scheme for mobile cloud computing service. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 869–876. IEEE, 2015. 18

Gaoqiang Zhuo, Qi Jia, Linke Guo, Ming Li, and Pan Li. Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016. 25

Justin Zobel and Alistair Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1): 18—-34, April 1998. 25