Electrical Engineering and Computer Science   College of Engineering and Computer Science

2012

Sensing as a service: A cloud computing system for mobile phone sensing

Xiang Xiang Sheng
Syracuse University, xsheng@syr.edu

Xuejie Xiao
Syracuse University, xxiao04@syr.edu

Jian Tang
Syracuse University, jtang02@syr.edu

Guoliang Xue
Arizona State University, xue@asu.edu

Recommended Citation

X. Sheng, X. Xiao, J. Tang, and G. Xue, " Sensing as a service: A cloud computing system for mobile phone sensing," in 11th IEEE SENSORS 2012 Conference, October 28, 2012 - October 31, 2012, Taipei, Taiwan, 2012.

Sensing as a Service: A Cloud Computing System for Mobile Phone Sensing

Xiang Sheng*, Xuejie Xiao*, Jian Tang* and Guoliang Xue†
*Department of Electrical Engineering and Computer Science, Syracuse University.
†School of Computing, Informatics, and Decision Systems Engineering, Arizona State University

Abstract—Sensors on (or attached to) mobile phones can enable attractive sensing applications in different domains such as environmental monitoring, social networking, healthcare, etc. We introduce a new concept, *Sensing-as-a-Service (S^2aaS)*, i.e., providing sensing services using mobile phones via a cloud computing system. An S^2aaS cloud should meet the following requirements: 1) It must be able to support various mobile phone sensing applications on different smartphone platforms. 2) It must be energy-efficient. 3) It must have effective incentive mechanisms that can be used to attract mobile users to participate in sensing activities. In this paper, we identify unique challenges of designing and implementing an S^2aaS cloud, review existing systems and methods, present viable solutions, and point out future research directions.

Index Terms—Mobile phone sensing, sensing as a service, cloud computing, energy-efficiency, incentive mechanisms.

I. INTRODUCTION

Most of current mobile phones (such as iPhone 4S, Samsung's Android phones, etc.) are equipped with a rich set of embedded sensors such as camera, GPS, WiFi/3G/4G radios, accelerometer, digital compass, gyroscope, microphone and so on. Moreover, external sensors can also be connected to a mobile phone via its Bluetooth interface. These sensors can enable attractive sensing applications in various domains such as environmental monitoring, social networking, healthcare, transportation, safety, etc. Mobile phone sensing has been studied by a few recent works [8], most of which, however, presented the design and implementation of a mobile phone sensing system for a particular application.

In this paper, we propose to leverage emerging cloud computing model to provide various sensing services using mobile phones for a large number of cloud users and introduce a new concept: *Sensing as a Service (S^2aaS)*. A typical S^2aaS cloud is illustrated in Fig. 1. In an S^2aaS cloud, multiple sensing servers (as shown in the figure) can be deployed to handle sensing requests from different locations. When a cloud user initiates a sensing request through an online form in a web server from either a mobile phone or a computer (desktop/laptop), the request will be forwarded to a sensing server which will then push the request to a subset of mobile phones that happen to be in the area of interest. The corresponding sensing task will be fulfilled by these mobile phones. The sensed data will then be collected by a sensing server, stored in the database and returned to the requester. An interesting feature of such a system is that a mobile phone
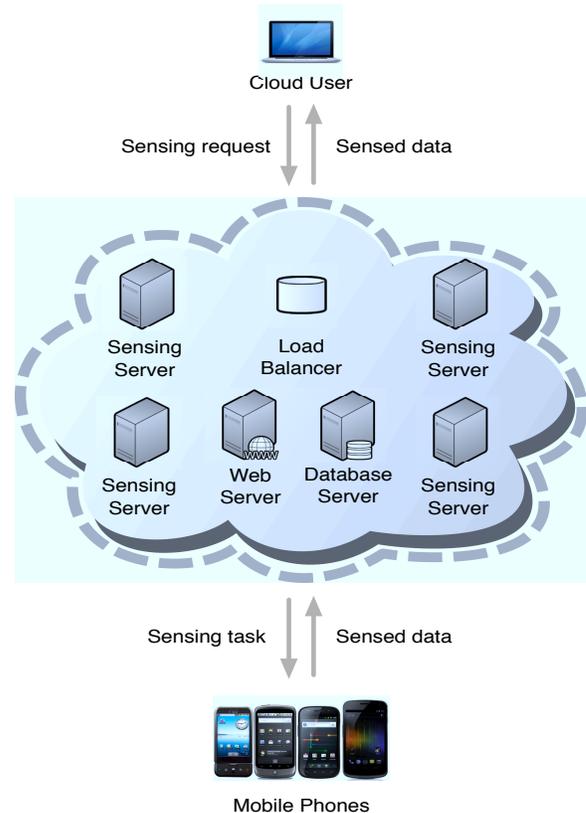
Fig. 1. An S^2aaS cloud

user (or simply mobile user) can be not only a cloud (service) user who can request sensing services from the cloud but also a service provider who fulfills sensing tasks according to sensing requests from other cloud users.

There are primarily two mobile phone sensing paradigms [8]: *Participatory Sensing and Opportunistic Sensing*. In participatory sensing, mobile users actively engage in sensing activities by manually determining how, when, what, and where to sense. In opportunistic sensing, sensing activities are fully automated without the involvement of mobile users. To provide sensing services for a large number of cloud users with different needs, an S^2aaS cloud must be able to support various participatory sensing and opportunistic sensing applications on different smartphone platforms. Performing sensing tasks may consume a significant amount of energy of a mobile phone. Therefore, without carefully managing very limited energy resources on mobile phones, users may end up with an awkward situation after performing a few sensing tasks, in which phones

are out of battery when they are needed to make phone calls. Moreover, most mobile phone sensing applications are location-dependent. If energy-hungry GPS is turned on during the whole sensing procedure, the battery may be drained very quickly. There is a large space for energy savings. However, fundamental energy-efficient resource management problems have not been well studied for mobile phone sensing. In addition, unlike a traditional sensor network which is usually operated by a single organization, mobile phones and their sensors are owned and controlled by different individual users. While participating in sensing activities, mobile users will consume their own resources such as battery and computing resources. More importantly, participating users will also expose themselves to potential privacy threats. Hence, a mobile user would not be interested in participating in mobile phone sensing, unless he/she receives a satisfying reward to compensate his/her resource consumption and potential privacy breach. A fundamental problem is how to provide incentives to attract these selfish mobile users to participate in sensing activities, which, to the best of our knowledge, has not been well addressed yet.

Developing a unified, green and incentive cloud computing system for mobile phone sensing is quite challenging. In this paper, we identify *unique challenges*, review existing systems and methods, present viable solutions, and point out future research directions. Even though sensed data processing and analysis, security and privacy are critical issues, they are out of scope of this work since they are common issues in sensor networks and mobile cloud computing systems, however, we aim to address research challenges unique to developing an S^2aaS cloud here.

The rest of the paper is organized as follows: We discuss general system design and implementation, energy-efficient sensing task management, and incentive mechanism design in Sections II, III and IV, respectively. The paper is then concluded in Section V.

II. SYSTEM DESIGN AND IMPLEMENTATION

The following important and special issues need to be carefully addressed for designing and implementing an S^2aaS: 1) The cloud system must be general enough such that it can support various opportunistic and participatory sensing applications (which may even involve a large variety of sensors), and there is very little overhead to launch a new sensing application/service on it. 2) New algorithms or policies that aim to improve the performance of the system can be easily and quickly deployed to replace the old inefficient ones. 3) Sensing energy consumption should be minimized such that mobile phones can undertake sensing tasks, and in the meanwhile, can still fulfill its regular duties, such as making phone calls, sending/receiving emails, browsing webpages, etc. 4) The system must have effective incentive mechanisms to attract mobile phone users to participate in sensing activities.

Recently, research efforts have been made to develop systems to support mobile phone sensing. In [5], Das *et al.* presented a Platform for Remote Sensing using Smartphones (PRISM), which allows application writers to package their applications as executable binaries and push them automatically to an appropriate set of phones. The Bubble-Sensing proposed in [10] allows sensing tasks to be posted at specific physical locations of interest. Cornelius *et al.* introduced AnonySense in [3], which is a sensor tasking and reporting system designed for both participatory and opportunistic sensing. Microblogs [7] is another system for participatory sensing, where users upload blogs annotated with sensed information (e.g., photos) to a micro-blog server. Mobile devices also upload their locations to the server periodically. However, these existing systems have the following problems: 1) PRISM [5] uses executable binaries to deliver sensing tasks to mobile phones, which is platform-dependent (Windows Mobile only) and may cause security issues. 2) AnonySense [3] uses a customized, yet very limitedly-used Lisp dialect for implementation. 3) Important issues, such as energy-efficiency and user incentives, have not been addressed in these related works.

The following functionalities should be supported by an S^2aaS cloud: (1) Web Interface: It needs to provide a web interface for cloud users, which can be accessed via a mobile phone or a regular computer. (2) Generating Sensing Tasks: It needs to generate new sensing tasks in a standard format based on request information collected from the web interface (e.g., what sensors to use, what data to collect, what is the area of interest, etc). (3) Recruiting Mobile Users: It needs to recruit a set of mobile phone users to participate in sensing activities for each incoming sensing task using an incentive mechanism (discussed in Section IV). (4) Scheduling Sensing Activities: It needs to schedule sensing activities of the set of mobile phones recruited for each sensing task using a given scheduling algorithm or policy (discussed in Section III) (5) Managing Sensors: an application needs to be deployed on each mobile phone to operate its sensors to perform the requested sensing actions, collect sensed data and send them to a sensing server. (6) Storing Data: It needs to store sensed data for future use. Essentially, currently available web servers (such as Apache HTTP server [1]) and database systems (such as BigTable [2]) can be used to provide the web interface and to store sensed data respectively. A sensing server needs to be developed to support functions (2)–(4). A mobile phone application needs to be developed to implement function (5).

To create a *unified* cloud computing system for mobile phone sensing, scripts written in a scripting language (rather than binary codes [5]) can be employed to describe every sensing task in a standard format and can then be pushed to mobile phones on which they will be executed with the help of an interpreter. Scripting languages can bring portability to the system such that the population of the sensing crowd can be effectively increased because sensing tasks described using scripts can run on hardware platforms with different CPU architectures, such as ARM, MIPS, SPARC, x86. Moreover, scripting languages can enable dynamic and flexible loading of programs on mobile phones because using a scripting language, an interpreter can be integrated into mobile applications to download and interpret the scripts on-the-fly, while, all binaries (packed as APK applications) need to be signed by Google before being loaded to users' mobile phones on the Android platform and a similar method is used on the iOS platform too. In addition, scripting languages can eliminate

potential security threats by running the scripts in a sandbox and only allowing them to use a *white* list of APIs such that they only interact with the hardware in the ways we trust.

Modular design and clearly defined interfaces will play a key role in supporting *configurability*. Every major functionality should be implemented as an independent module with well-defined interfaces to interact with other components. In order to improve *energy-efficiency*, efficient and practical algorithms need to be developed for mobile phone scheduling on the server side as well as sensing task scheduling on the mobile client side with the objective of minimizing and balancing energy consumption. This will be discussed in greater details in Section III. Furthermore, game-theoretic incentive mechanisms need to be developed for attracting user participation, which will be discussed in Section IV.

III. Energy-Efficient Sensing Task Management

Energy-efficiency issues have been studied in the context of mobile phone sensing recently [10], [12], [13], [22]. In [10], the authors presented the design, implementation and evaluation of the Jigsaw continuous sensing engine for mobile phones, which balances performance needs and resource demands. The authors of [12] presented the design, implementation and evaluation of several techniques to optimize the information uploading process for continuous sensing on mobile phones. Energy-efficient GPS-based location sensing methods were presented in [13], [22]. However, most such related works were focused on a single mobile phone. We aim to minimize energy consumption via a collaborative sensing approach in which the cloud is used for coordinating sensing activities of multiple mobile phones.

Only few recent works addressed collaborative sensing with mobile phones. In [11], the authors presented analytical results on the rate of information reporting by uncontrolled mobile sensors needed to cover a geographical area. In [18], the authors introduced mechanisms for automated mapping of urban areas, which provide a virtual sensor abstraction to applications. They also proposed spatial and temporal coverage metrics for measuring the quality of sensed data. In [17], the authors proposed the Aquiba protocol, which exploits opportunistic collaboration of pedestrians and evaluate its performance via simulations.

Centralized and distributed collaborative sensing algorithms have been proposed in [15], [20], [21] to address different coverage and connectivity problems in mobile sensor networks (where sensor mobility can be controlled to achieve certain sensing coverage). Specifically, In [21], Zhou *et al.* presented a dynamic programming based algorithm to determine how to deploy mobile sensors in a sensor network to enhance its connectivity and coverage. Distributed GPS-less algorithms were presented for a sensing coverage problem in [20]. In [15], Saipulla *et al.* explored the fundamental limits of sensor mobility on barrier coverage and presented a sensor mobility scheme that constructs the maximum number of barriers with the minimum sensor moving distance. However, the algorithms presented in these works cannot be applied here because the mobility of mobile phones is usually uncontrollable. Research on coverage and scheduling for mobile phone sensing is still in its infancy.

We need to consider the following optimization problem (which has not been well addressed yet): Given a set of target points or a target region, a set of mobile phones and a deadline, find a sensing schedule (which specifies when to sense for each mobile phone) such that the total energy consumption is minimized subject to a coverage constraint. In a recent work [16], under the assumption that the moving trajectory of each mobile user is known in advance, a polynomial-time algorithm was presented to obtain minimum energy sensing schedules that can ensure full coverage of given roadways. Moreover, the authors addressed individual energy consumption and fairness by presenting an algorithm to find fair energy-efficient sensing schedules. It has been shown by simulation results based on real energy consumption and location data that compared to traditional sensing without collaborations, collaborative sensing achieves over 80% power savings. Even though these algorithms can produce optimal solutions, *practical* algorithms need to be developed for these mobile phone scheduling problems without assuming the mobility pattern of each mobile phone user is known beforehand. In addition, GPS is energy-hungry and keeping GPS on during the whole sensing procedure is not feasible since it may drain a phone battery quickly. Other approaches, such as WiFi or cellular signals, can also be used to obtain location information, which consume much less energy but provide less accuracy. Hence, GPS-less algorithms are needed for sensing scheduling. The scheduling problems become very challenging without accurate location information. First, a probabilistic coverage model needs to be developed to calculate the probability that a target point (or area) is covered if a mobile phone is scheduled to sense at a location which it believes to be (x, y) (The actual location may not be (x, y)), and the coverage probability given by a sensing schedule. Second, a simple and practical method is needed to predict the mobility of mobile users based on historical data. Moreover, efficient algorithms (based on the coverage model and the mobility prediction algorithm) need to be designed to solve the scheduling problems.

A sensing task will be assigned to multiple mobile phones. Correspondingly, a mobile phone may be used to process multiple sensing tasks. Hence, sensing task scheduling algorithms are also needed to schedule multiple sensing tasks on a mobile phone. The following optimization problem needs to be addressed: given a set of sensing tasks (on a mobile phone), each with certain temporal requirement (i.e., must be completed at a particular time or during a certain period), spatial requirement (i.e., must be performed at a particular location or in a certain area), or both, find a schedule with minimum energy consumption for performing these tasks such that the given requirements are met. To the best of our knowledge, this problem has not been well studied before. One trivial solution is to treat each sensing task as an independent task and handle them one by one. However, this may not be energy-efficient because multiple sensing tasks may share one or multiple sensing actions (e.g., request location information from GPS). The best way may be to group multiple correlated tasks together by exploiting the temporal-spatial correlations

between them, schedule sensing actions associated with them and determine when to conduct common sensing actions based on user mobility status with the objective of minimizing energy consumption and satisfying the temporal and spatial requirements. Note that scheduling problems discussed in this section are only related to opportunistic sensing.

IV. INCENTIVE MECHANISM DESIGN

There are few research studies on the incentive mechanism design for mobile phone sensing. In [14], Reddy *et al.* developed recruitment frameworks to enable the system to identify well-suited participants for sensing services. However, they focused only on user selection, not incentive mechanism design. In [4], Danezis *et al.* developed a sealed-bid second-price auction to motivate user participation. However, the utility of the platform was neglected in the design of auction. In [9], Lee and Hoh designed and evaluated a reverse auction based dynamic price incentive mechanism, where users can sell their sensed data to the service provider with users' claimed bid prices. However, the authors did consider truthfulness, which is an important property. In a recent paper [6], the authors analyzed and compared different incentive mechanisms for a client to motivate the collaboration of smartphone users on both data acquisition and distributed computing applications. However, they formulated their problems based on a simple model without carefully addressing users' contributions.

In [19], Yang *et al.* proposed two models for incentive mechanisms: platform-centric and user-centric. In the platform-centric model, the cloud computing platform has the absolute control over the total payment made to mobile users, and users can only tailor their actions to cater for the platform. Whereas in a user-centric incentive mechanism, the roles of the platform and users are reversed. To assure himself/herself of the bottom-line benefit, each mobile user announces a reserve price, the smallest price at which he/she is willing to sell a service. The platform then selects a subset of mobile users and pay each of them an amount that is no smaller than the user's reserve price. In the platform-centric model, a user participating in mobile phone sensing will earn a payment that is no lower than its cost. However, it needs to compete with other users for a fixed total payment. In the user-centric model, each user asks for a price for its service. If selected, the user will receive a payment that is no lower than its asked price. Unlike the platform-centric model, the total payment is not fixed for the user-centric model. Hence, the users have more control over the payment.

Based on some simple utility functions, the authors of [19] presented a Stackelberg Game based approach for the platform-centric model to compute a Stackelberg equilibrium, which maximizes the utility of the platform, and ensures that no user has the incentive to change his/her strategy. They also designed an auction-based incentive mechanism for the user-centric model, which is computationally efficient, individually-rational, profitable and truthful. The design of incentive mechanisms with desirable economic properties for more realistic utility functions is an important topic for future research.

V. CONCLUSIONS

In this paper, we introduced a new concept, Sensing as a Service (S^2aaS), and identified unique challenges of developing an S^2aaS cloud, which include: 1) support for various sensing applications; 2) energy-efficiency; 3) incentive mechanism design. We then reviewed existing systems and methods, presented viable solutions, and pointed out future research directions. Specifically, we described the basic functionalities that an S^2aaS cloud needs to have and proposed to use scripts to describe various sensing tasks and enable secure and flexible loading of them over different smartphone platforms. Moreover, we introduced energy-efficient sensing scheduling problems and pointed out the right directions for developing effective scheduling algorithms. In addition, we discussed two models for incentive mechanism design, platform-centric model and user-centric model, and described the desirable properties for incentive mechanisms under these two models.

REFERENCES

[1] Apache HTTP Server, *http://httpd.apache.org/*
[2] BigTable, *http://en.wikipedia.org/wiki/BigTable*
[3] C. Cornelius, *et al.* , Anonysense: privacy-aware people-centric sensing, *ACM MobiSys'2008*, pp. 211–224.
[4] G. Danezis, S. Lewis and R. Anderson, How much is location privacy worth? *WEIS'2005*.
[5] T. Das, *et al.* , PRISM: platform for remote sensing using smartphones, *ACM MobiSys'2010*, pp. 63–76.
[6] L. Duan, *et al.* , Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing, *IEEE Infocom'2012*, pp. 1701–1709.
[7] S. Gaonkar, *et al.* , Micro-blog: sharing and querying content through mobile phones and social participation, *ACM MobiSys'2008*, pp. 174–186.
[8] N. D. Lane *et al.* , A survey of mobile phone sensing, *IEEE Communications Magazine*, Vol. 48, No. 9, 2010, pp. 140–150.
[9] J. Lee, and B. Hoh, Sell your experiences: a market mechanism based incentive for participatory sensing, *IEEE PerCom'2010*, pp.–60–68.
[10] H. Lu, *et al.* , Bubble-sensing: Binding sensing tasks to the physical world, *Journal of Pervasive and Mobile Computing*, Vol. 6, No. 1, 2010, pp. 58–71.
[11] S. Madhani, M. Tauil and Z. Tao, Collaborative sensing using uncontrolled mobile devices, *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2005.
[12] M. Musolesi, *et al.* , Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones, *Pervasive Computing, Lecture Notes in Computer Science*, Vol. 6030, 2010, pp. 355–372.
[13] J. Paek, J. Kim and R. Govindan, Energy-efficient rate-adaptive GPS-based positioning for smartphones, *ACM MobiSys'2010*, pp. 299–314.
[14] S. Reddy, D. Estrin and M. B. Srivastava, Recruitment framework for participatory sensing data collections. *Pervasive'2010*, pp. 138–155.
[15] A. Saipulla, *et al.* , Barrier coverage with sensors of limited mobility, *ACM Mobihoc'2010*, pp. 201–210.
[16] X. Sheng, J. Tang and W. Zhang, Energy-efficient collaborative sensing with mobile phones, *IEEE Infocom'2012*, pp. 1916–1924 .
[17] N. Thepvilojanapong, *et al.* , Opportunistic collaboration in participatory sensing environments, *ACM MobiArch'2010*, pp. 39–44.
[18] H. Weinschrott, *et al.* , StreamShaper: coordination algorithms for participatory mobile urban sensing, *IEEE MASS'2010*, pp. 195–204.
[19] D. Yang, G. Xue, X. Fang, and J. Tang, Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing, *MobiCom'2012*, Accepted.
[20] C. W. Yu, *et al.* , Coverage algorithms in GPS-less wireless mobile sensor networks, *ACM Mobility'2008*.
[21] S. Zhou, M. Y. Wu, and W. Shu, Finding optimal placements for mobile sensors: wireless sensor network topology adjustment, *IEEE Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, 2004, pp. 529–532,
[22] Z. Zhuang, K-H, Kim and J. P. Singh, Improving energy efficiency of location sensing on smartphones, *ACM MobiSys'2010*, pp. 315–330.