Energy Saving Techniques in Mobile Crowd Sensing: Current State and Future Opportunities

Jiangtao Wang[#], Yasha Wang[#], Daqing Zhang, Sumi Helal
 # Corresponding author

Abstract: With the prevalence of sensor-rich smartphones, Mobile Crowd Sensing (MCS) is becoming an emerging paradigm to perform urban sensing tasks in recent years. In MCS systems, it is important to minimize the energy consumption on devices of mobile users, as high energy consumption severely reduce their participation willingness. In this article, we provide a comprehensive review of energy saving techniques in MCS and identify future research opportunities. Specifically, we analyze the main reasons for energy consumption in MCS and present a general energy saving framework named ESCrowd that we use to describe the different detailed MCS energy saving techniques. We further present how the various energy saving techniques are utilized and adopted within MCS applications and point out to their existing limitations which inform and guide future research directions.

1. Introduction

Urban sensing is crucial in collecting real-time information and extracting community intelligence in a city, including environment information (e.g. air quality, noise), infrastructure status (e.g., missing manholes, broken streetlights) and city dynamics (e.g., traffic congestions, the flow of people), etc. Traditional urban sensing systems usually rely on specialized infrastructure (e.g., air quality monitoring stations, surveillance cameras), which requires a high cost in deployment and maintenance. With the prevalence of sensor-rich smartphones, *Mobile Crowd Sensing* (*MCS*) [1] becomes a promising paradigm, which leverages the mobility of mobile users, the sensors embedded in mobile phones and the existing communication infrastructure to accomplish urban sensing tasks. Compared to traditional infrastructure-based approaches, MCS can sense large urban regions less costly and more efficiently.

As Fig.1 shows, in an MCS platform, there are mainly two roles: MCS *organizer* (or requester) who is the person or organization publishing, managing and coordinating the sensing task, and MCS *workers* (or participants) who are the mobile users collect and report sensing data through their mobile devices (e.g., mobile phone). The success of MCS depends on if the organizer can recruit large numbers of smartphone users as workers to collect mobile sensor data. One of the major concerns of the workers is the energy consumption when completing the MCS tasks, which is caused by the raw sensor data collection, local data analytics and data transmission to the cloud server. Energy consumption has a direct impact on the battery life of a worker's smartphone. If the energy consumption of an MCS task is too high, it will severely reduce the mobile users' willingness of becoming a crowd worker. Therefore, keeping the energy consumption burden placed on workers as low as possible is critical to the success of MCS [1,2].

In this article, we specifically focus on the energy saving techniques in MCS systems and provide a comprehensive review with future research opportunity. We first identify main reasons for energy consumption in MCS with key insights or observations. Then, we present the general energy saving technical framework by organizing the state-of-the-art studies according to different stages of MCS. Subsequently, detailed techniques in the above framework are introduced with the case study of some typical MCS applications. Finally, we list the research gaps with future research directions.

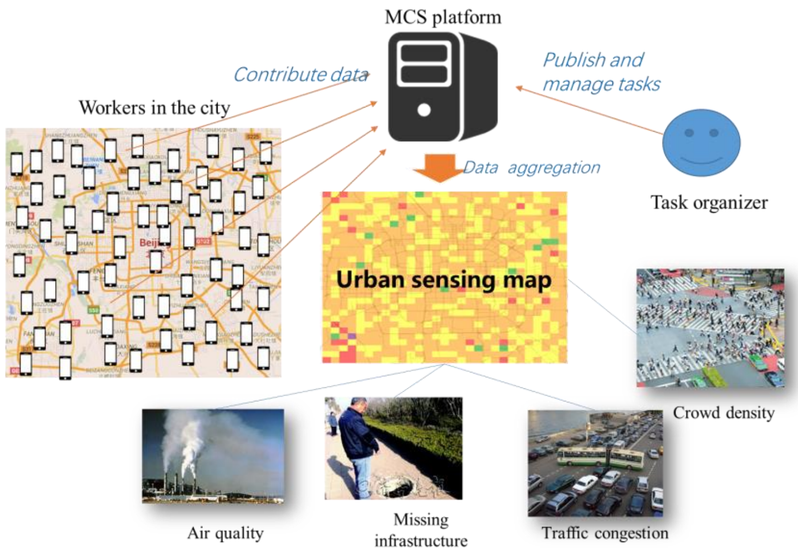

Fig.1. Mobile crowd sensing: the basic idea and main roles

2. Energy Consumption in MCS

In this section, we first provide the preliminary for general MCS applications, and then present an anatomy of the energy consumption issues in MCS applications.

2.1 Preliminary for Mobile Crowd Sensing

In Fig.2, we present the typical functioning components of MCS applications, which follows the client-and-server architecture. In the client side (i.e., the phone side), raw sensor data are collected on devices and processed by local analytic algorithms to produce consumable data for applications. In the server side, the consumable data from multiple workers' devices are sent to the cloud server for aggregation and mining.

We take the MCS-based queue time estimation application named CrowdQTE [3] as an example to illustrate the above functionalities and architecture. CrowdQTE utilizes the sensor-enhanced mobile phone to monitor and provide real-time queue time information for various queuing locations. When people are waiting in a line, the phone side utilizes the accelerometer sensor data and ambient contexts to automatically detect the queueing behavior and calculate the queue time. In the server side, CrowdQTE organizes data from different phones in a given place (e.g., a specific supermarket) into data groups according to the sensing time, then eliminates noisy data from each group, and at last calculates the average of valid queue time/condition as the estimation.

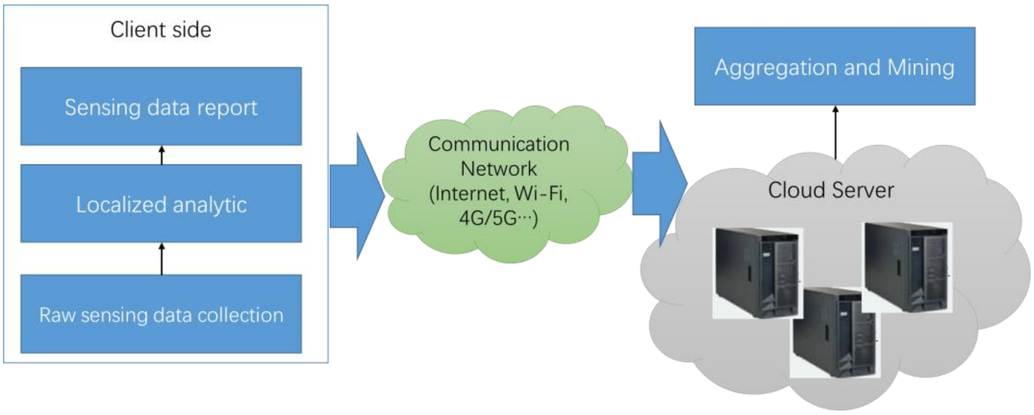

Fig.2 The architecture and main functioning components for an MCS application

2.2 Energy Consumption in MCS: An Anatomy

Finding main reasons for energy consumption in MCS and analyzing relevant characteristics helps us design appropriate strategies for energy saving. Thus, in this section, we present an anatomy for the energy consumption[1] in MCS with technical challenges. Keeping the functionality of MCS system's mobile phone side in mind, the energy is mainly consumed in the following aspects.

- ***Raw Sensing Data Collection Consumption.*** Raw sensor data sampling is a major source of energy consumption in MCS systems. First, the energy efficiency varies dramatically from one type of sensor to another. This fact indicates that adopting more energy-efficient sensors can reduce the energy consumption in MCS data collection. Second, even for the same type of sensor, the data sampling rate has a big impact on the energy consumption. For example, different from general crowdsourcing tasks, MCS requires the workers' physical presence at certain locations so that the continuous tracking of workers' location always leads to high energy consumption. However, if we can adjust the sampling rate of localization information appropriately (e.g., only collect localization data when the workers are moving), the mobile phone's energy consumption is significantly reduced. In summary, *it is important to better control the energy consumption in data collection by taking both the type of sensors and sampling rate into account*.

- ***Localized Analytic Consumption***. In many MCS applications, the raw sensing data should be processed in smartphones to infer higher-level contexts, which is called local analytic in this article. The local analytic sometimes brings considerable energy consumption to the mobile phone. For example, in an MCS-based queue time estimation system [3], various types of raw sensor data (e.g., accelerometer data, GPS locations, and acoustic data) are fused with complicated and energy-consuming algorithms to estimate the queuing status then calculate the current queue time of a certain location. Therefore, *it is a challenge in designing an optimized local analytic mechanism to minimize the energy consumption while ensuring the basic functionality of the MCS application*.

---

[1] Note that in this article we only consider the energy consumption issue for workers' mobile phone without considering that for the cloud server.

- ***Sensing Data Report Consumption.*** In MCS systems, the sensing data transferred from the mobile phone to the cloud server also serves as a major source of energy consumption. There are two key insights as follows. First, the type of communication infrastructures that an MCS system chooses is an important factor. Fig.3 shows energy consumption comparison of various communication infrastructures. For example, transferring data via the 4G/5G network is much more energy-consuming than others such as Wi-Fi and Bluetooth. However, these energy-efficient networks may not always available. Meanwhile, some MCS tasks require real-time data uploading. Therefore, *it is a challenge in switching between different networks by taking both the application requirement and energy efficiency into account*. Second, uploading data to the server with some opportunities (e.g., when users place the phone call or use applications) can significantly save energy consumption in data transferring. Thus, *how to detect and take advantage of such opportunities is an important research issue*.

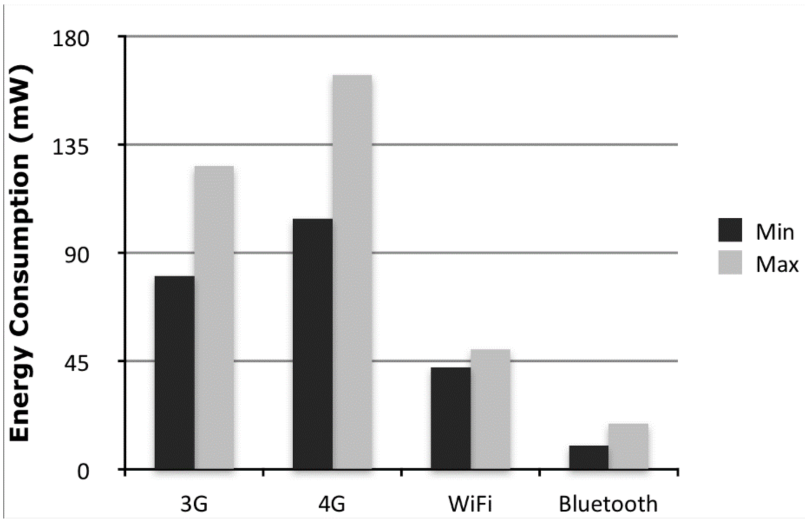

Fig.3 Relative view of energy consumption comparison for various communication infrastructures: minimum and maximum measurements.

3. Energy Saving in MCS: Technical Framework

The lifecycle of MCS consists of four stages, including sensing task assignments, sensing data acquisition & inference, sensing data transferring, and sensing data aggregation. In this section, we present the overview of state-of-the-art energy saving techniques as a general framework, named ESCrowd, which is organized in the perspective of different stages in MCS (see Fig.4). In the ESCrowd framework, corresponding techniques are adopted and integrated in each stage to reduce the energy consumption. Besides, Fig.4 also illustrates the types of energy consumptions (presented in section 2) that each technique can reduce (i.e., the arrow in Fig.4 liking the technique and the type of consumption).

- **Energy-aware task assignment.** Reducing the number of recruited workers can save the overall energy consumption in MCS. Thus, in the task assignment phase, we can select a minimal number of workers while ensuring a predefined sensing quality [4,5]. As Fig.4 shows, the optimized task assignment approach will reduce overall energy consumption in MCS with less collected and uploaded data (i.e., reducing total sensing data collection consumption, total local analytics consumption, and total sensing data report consumption).

- **Energy-aware data acquisition & inference.** For the raw sensing data collection, the energy consumption can be saved in the following ways: 1) utilizing energy-efficient sensors to replace traditional sensors that are more energy consuming [8]; 2) dynamically adjusting the sampling frequency [9,10]. For the local knowledge inference, techniques such as code offloading or remote execution [9] can be utilized to reduce the energy consumption in the smartphone.

- **Energy-aware data transferring.** To reduce the energy consumption in sensing data report, there are commonly two ways: 1) using relatively low-power wireless networks [12,13]; 2) catching the opportunities to upload data when the mobile users place phone calls or use mobile applications [14].

- **Energy-aware data aggregation.** The high spatio-temporal correlations exist in most urban data, e.g., air quality and noise, which provides the basis for high-quality missing data inference. Thus, we can select only a small portion of the target area for sensing while inferring the data of the remaining unsensed area with high accuracy on the server [15]. This strategy will reduce total energy consumption in the smartphone by minimizing the volume of data that needed to be collected and reported (i.e., reducing total sensing data collection consumption and total sensing data report consumption).

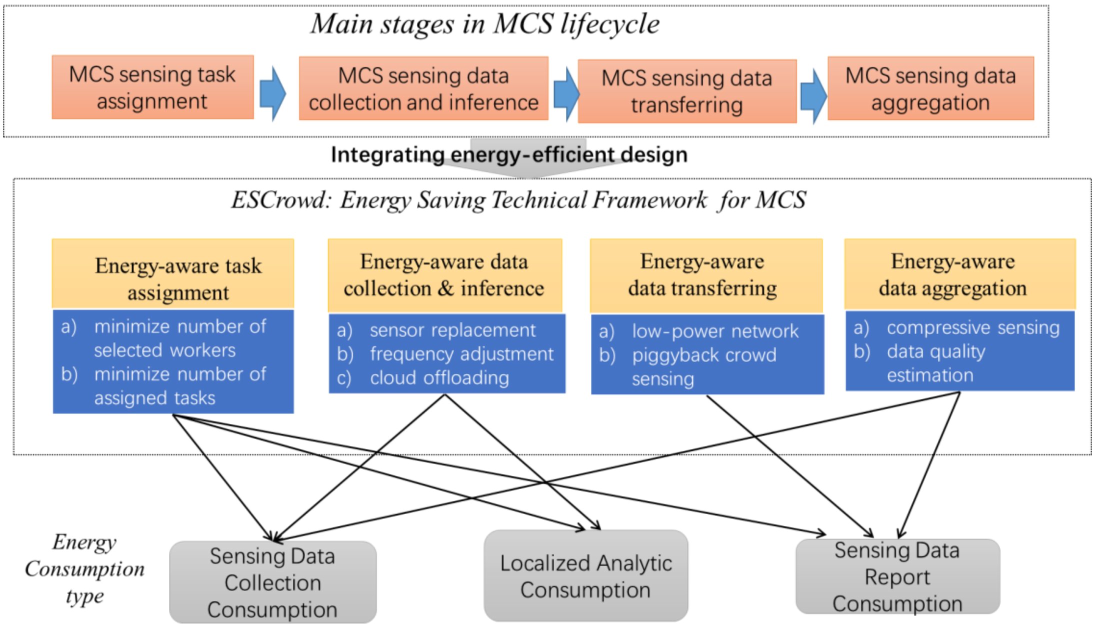

Fig.4 ESCrowd framework overview

According Fig.4, the energy saving techniques in four stages discussed in the paper are independent of each other. However, as implied in Fig.4, energy saving techniques in different stages and the type of saved energy are correlated with each other. For example, by using the energy-aware task assignment techniques, the total number of assigned tasks is minimized. In this case, both three types of energy consumption are saved. In contrast, the energy-aware data transferring only saves the sensing data report consumption.

4. Detailed Techniques

Section 3 presented the general technical framework for energy saving in MCS. In this section, we present some typical studies for each type of technique in this framework with more details.

4.1 Energy-aware task assignment.

In order to minimize the overall energy consumption of an MCS task, the research objective becomes keeping the energy consumption of each mobile device low and finding the minimal number of workers while ensuring a predefined sensing quality. For instance, to minimize the energy consumption, [4] proposed a framework to select a minimum number of workers while ensuring the required spatial-temporal coverage. The authors in [5] formulated a task allocation problem, whose objective is to maximizing k-coverage quality while minimizing energy consumption in MCS task allocation. The authors in [6] formulated another MCS task allocation problem, in which the objective is to maximize the task quality by given the limited overall energy consumption. The study in [7] develops a novel task allocation algorithm by considering the energy consumption, worker' reputation, and budget limitation.

The above energy-efficient MCS task assignment issues can be formulated as combinatorial optimization problems, which attempt to find an optimal solution from a large search space. Intuitively, it is easy to think of a brute-force approach, where it can estimate the utility of each possible combination so that the optimal one can be obtained. However, the formulated combinatorial optimization problems are usually NP-hard, thus the brute force approach is not acceptable when there are a large number of workers or tasks. Therefore, existing research works commonly design approximation allocation algorithms to achieve the near-optimal solution, and the general process for energy-aware MCS task allocation consists of two main components. (1) Utility Estimation: with the consideration of task quality and overall energy consumption, this component is for estimating the utility of a given set of workers. Usually, the estimation needs the understanding of the workers' mobility pattern so that the historical mobility records profiling and mobility prediction are the basic components. (2) Searching Process: the heuristic searching algorithms are adopted to obtain a near-optimal solution.

4.2 Energy-aware data acquisition & inference

For the raw sensing data collection, the energy consumption is primarily incurred by the sensor itself. One possible way is to design novel method to use energy-efficient sensors to replace traditional sensors that are more energy to consume. For example, the authors in [8] introduce a novel sensing approach which lowers the power requirement for motion sensing by orders of magnitude. The key idea is that it uses an ultra-low-power method for passively sensing body motion using static electric fields to replace the traditional accelerometer-based methods. Besides, dynamically adjusting the sensing frequency (sampling rate) is another important way to reduce the energy consumption in data acquisition. Several studies have been proposed to dynamically adjust the sampling frequency to conserve power based on either the battery level or users' movements. For example, the proposed methods in [9] detect users' movements dynamically and adjust the sampling frequency accordingly. The key idea behind is that it is inefficient to keep collecting users' location information when their locations have not changed. Thus, at the time when devices are stationary, the corresponding sensors should be shut down to save power until any motion is detected. The systems proposed in [10] consider how to schedule heterogeneous sensors (e.g., accelerometer, WiFi, and cell towers) to better manage the tradeoff between the energy efficiency and sensing data quality.

For the local knowledge inference, techniques such as cloud offloading [11] can be utilized to reduce the energy consumption in the smartphone. For example, observing that in many sensing scenarios the location information can be post-processed when the data is uploaded to a server, the authors in [11] design a cloud-offloaded solution that allows a sensing device to aggressively duty-cycle its GPS receiver and log just enough raw GPS signal for post-processing.

4.3 Energy-aware data transferring.

To reduce the energy consumption in data uploading, on way is to use relatively low-power wireless networks methods, such as WiFi, to transfer data, instead of 3G/4G. The authors in [12] propose an MCS framework, in which data collection mainly depends on the data transmission among mobile workers via Bluetooth or WiFi, which significantly save the energy consumption. However, these networks may not always available, relying on the opportunities to connect with them often leads to delay. Therefore, the authors in [13] propose an MCS framework, named effSense. effSense adopts a distributed decision-making scheme to determine the timing and type of network to upload data, which consider both the delay tolerance and energy consumption.

Another alternative for uploading data more energy-efficiently is catching the opportunities when the mobile users place phone calls or use mobile applications. One typical work is the Piggyback Crowd Sensing (PCS) [14], a system for collecting mobile sensor data from smartphones that lower the energy overhead of user participation. Their approach is to collect sensor data by exploiting Smartphone App Opportunities, that is, those times when smartphone users place phone calls or use applications. For example, [14] shows the energy consumption of microphone and GPS sensing with and without app usage at the same time, in which performing sensing while an app is in use requires 43% less CPU-related energy during the sampling operation. The reason why the energy cost of sensing can be significantly reduced at these times is that the required smartphone components (e.g., CPU or even the sensor itself) are already activated from an idle state. To efficiently use these sporadic opportunities, PCS builds a lightweight, user-specific prediction model of smartphone app usage, which is used to drive a decision engine considering the expected energy/quality trade-offs.

4.4 Energy-aware data aggregation.

Inference techniques can be used in aggregation phase to infer missing data from the collected. In this way, it will minimize the volume of data that needed to be collected and reported, which significantly reduce the total energy consumption in MCS. The authors in [15] propose a sparse MCS framework, which intelligently selects only a small portion of the target area for sensing while inferring the data of the remaining unsensed area with high accuracy on the server. Note that the data aggregation has to be considered together with the task assignment, and the following issues are addressed in [15]: 1) Missing-Data Inference: how to infer the missing data of the unsensed cells with high accuracy? 2) Optimal Task Assignment: how to select a minimum number of spatio-temporal cells for task assignment while ensuring the inferred data quality? To address the missing-data inference issue, the authors in [15] transform it into a matrix completion problem and adopt compressive sensing based approach to solve it. For the optimal task allocation, it allocates the task to the cells that has the highest uncertainty (e.g., largest variance on these inferred values).

5. Energy Saving in Typical MCS Applications

As this article gives a taxonomy of the various energy saving techniques, it is interesting to see how these techniques can are used be in various MCS applications or frameworks, and Table 1 presents such a comparison (Due to the limit in archival references, we give the name and sensing target description of the MCS application/framework, so that the readers can easily find the references online).

First, we can see from Table 1 that the most widely adopted technique is the energy-aware data acquisition & inference (e.g., dynamically adjusting the sampling rate or using the low-power sensors) and data transferring (e.g., switching to the low-power network), and such techniques can be considered for different sensing targets. In fact, no matter what the sensing target is, MCS applications usually require the continuous reporting of mobile users' location information, so that adjusting the sampling rate or using the low-power localization sensor should be one of the commonly adopted mechanisms. We still take the queue time estimation system CrowdQTE [3] as an example. In order to determine if a mobile user is currently located in a certain place, the phone side should continuously track each potential worker's location, which will be quite energy-consuming. Therefore, CrowdQTE adopts the following mechanism. First, the system collects and stores the cell ID of each cellular tower in which the POI (point-of-interest) is located, and the geographical position of each cell tower. The phone-side application gets the current cell ID and calculates the distance between the current location and the target POIs. When the current location is far away (e.g., more than 5 KM) from any of the target POIs, the system conducts the above operation every 15 minutes. Otherwise, the cell ID based localization will be operated every 5 minutes. Second, when the system detects that the participant is near a POI, the localization switches to the Wi-Fi based approach. The system will calculate the similarity between the Wi-Fi fingerprint between current location and the target POI in this cell. If the similarity is less than a threshold, then it considers the participant is inside a specific POI.

Second, we also notice from Table 1 that the energy-aware data aggregation is usually applied in air quality or temperature sensing, because the spatio-temporal correlation of such urban sensing data enables the missing data inference. Besides, energy-aware data aggregation and task assignments are usually jointly adopted, in which the task assignments collect the most informative sensing data and the aggregation phase infers the missing data based on the collected one. For example, CCS-TA [15] is MCS-based system which aim collect real-time temperature and air quality measurement in a large city. In this system, the city is divided into M subareas (e.g., 1km*1km per subarea) and the entire sensing period is divided into N equal-length cycles (e.g., one hour per cycle), thus a total of M*N spatial-temporal cells are constructed. The goal of CCS-TA is to obtain accuracy-guaranteed sensing data in each spatial-temporal cell. If it recruits worker to collect sensing data for each cell, the total energy consumption for the workers would be very high as the city is very large. In this case, it only collects the sensing data in the most "informative" cells and deduce the data in the rest ones where the sensor reading are not collected.

Table 1 Comparison of Some Typical MCS Applications in Energy Saving

MCS app & sensing target	**Energy-aware task assignment**	**Energy-aware data acquisition & inference**	**Energy-aware data transferring**	**Energy-aware data aggregation**
PEIR (air quality)		★		
BikeNet(air quality)		★	★	
CrowdRecruiter (air quality)	★		★	
U-Air (air quality)	★			★
CCS-TA (temperature)	★			★
Nericell (road condition)		★	★	
CrowdQTE(queue time)		★		
LineKing(queue time)		★		
EarPhone(noise level)		★	★	
SoundofTheCity(noise level)		★	★	
Map++(semantic labeling)		★		

6. Future Research Opportunities For Energy-efficient MCS

Existing work has proposed various techniques to better control the energy consumption in MCS. We next highlight several directions for future research.

- **Jointly considering overall and individual consumption**. The existing techniques for energy saving in MCS can be divided into two categories based on the objective: the Energy-aware task assignment and data aggregation aims at reducing the overall energy consumption of an MCS task, while the energy-aware data collection & inference and energy-aware data transferring aims at reducing the energy consumption for an individual crowd worker. However, to the best of our knowledge, the state-of-the-art research work studies the overall and individual energy saving issues separately. In real-world application scenarios, such separation has shortcomings. For example, it is not good task assignment solution if the overall energy consumption is minimized but some individual workers are assigned with too many tasks. In this case, the battery life of the overloaded workers' mobile devices would become much shorter, which is likely to discourage their participation willingness or even make them quit the MCS task. We should jointly consider both the overall consumption and the load balance in individuals, together with other factors in MCS (such as coverage requirement, budget limitation, delay tolerance, etc.).

- **Handling opposing factors in energy saving**. Existing work has proposed various techniques to keep the energy consumption in MCS as low as possible. However, there are some conflicting factors that need to be more carefully considered, that is, reducing one type of consumption may bring the extra consumption of another type. For example, there exists a conflict between the local analytics and data transferring, because offloading the code of local analytics brings the extra consumption in data transferring. In order to achieve the optimal energy efficiency, determining which part of the analytic process should be offloaded and which should not is the key research challenge.

- **Introducing more personalized mechanisms.** Different workers have different preference or attitude towards the energy consumption issue. For example, some workers may prefer to charge their phone frequently in order to get more incentive rewards, while others only want to charge one time a day. Besides, the remaining battery is also an important factor with personalized patterns, because the energy would be consumed more severely when the remaining battery level is low. Thus, we need to learn the worker's historical behavior in phone usage and predict the remaining battery level in future sensing cycles, which would be helpful in worker selection problems. For example, if we can accurately predict that the remaining batter of worker *u* in a sensing cycle (e.g., 6:00pm~7:00 pm), the task assignment algorithm would reduce the probability to select worker *u* in this cycle.

- **Sharing sensing data among multiple tasks.** The objective of existing work is common to minimize the energy consumption of overall task or individual worker's mobile device. However, with the popularity of MCS paradigm, the same worker usually undertakes multiple types of sensing task. Although the sensing phenomenon among these tasks is quite different, the required set of sensors usually has an intersection. For example, the queue time estimation task in [3] needs to use GPS, accelerometers, and microphones, while noise level monitoring task requires GPS and microphones. In this case, the GPS and accelerometers can be shared. Therefore, intuitively, sharing sensing data among multiple tasks can reduce the energy consumption. However, as different tasks have different requirements data quality and delay tolerance, how to jointly optimize multiple tasks with the energy consumption issue in mind is a research challenge (e.g., how to collaboratively schedule heterogeneous kinds of sensors).

7. Conclusion

In this article, we present a survey of energy saving techniques in mobile crowd sensing. Specifically, by organizing the state-of-the-art work in the perspective of the stages in MCS, we present a general energy saving techniques framework named ESCrowd. According to ESCrowd, various kinds of MCS energy saving strategies are introduced in details. In the end, we point out some future research directions that may make the MCS more energy-efficient.

ACKNOWLEDGEMENT

We really appreciate all the reviewers and editors for their insightful comments, which significantly improve the paper. This work was mainly funded by NSFC Grant (No. 61702017).

References

1. R.K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. IEEE Communications Magazine, 49:32–39, 2011.
2. Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N. Y., Huang, R., & Zhou, X. (2015). Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. ACM Computing Surveys (CSUR), 48(1), 7.
3. Wang, J., Wang, Y., Zhang, D., Wang, L., Chen, C., Lee, J. W., & He, Y. (2016). Real-time and generic queue time estimation based on mobile crowdsensing. Frontiers of Computer Science, 2017.11(1), 49-60.

4. Zhang, D., Xiong, H., Wang, L., & Chen, G. (2014, September). CrowdRecruiter: selecting workers for piggyback crowdsensing under probabilistic coverage constraint. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 703-714). ACM.
5. Xiong, H., Zhang, D., Chen, G., Wang, L., Gauthier, V., & Barnes, L. E. (2016). iCrowd: Near-optimal task allocation for piggyback crowdsensing. IEEE Transactions on Mobile Computing, 15(8), 2010-2022.
6. Liu, C. H., Zhang, B., Su, X., Ma, J., Wang, W., & Leung, K. K. (2015). Energy-aware worker selection for smartphone-enabled mobile crowd sensing. IEEE Systems Journal, Volume: PP, Issue: 99, 1-12.
7. Wang, W., Gao, H., Liu, C. H., & Leung, K. K. (2016). Credible and energy-aware worker selection with limited task budget for mobile crowd sensing. Ad Hoc Networks, 43, 56-70.
8. Cohn, G., Gupta, S., Lee, T. J., Morris, D., Smith, J. R., Reynolds, M. S., & Patel, S. N. (2012, September). An ultra-low-power human body motion sensor using static electric field sensing. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (pp. 99-102). ACM.
9. Prelipcean, A. C., Gidófalvi, G., & Susilo, Y. O. (2014). Mobility collector. Journal of Location Based Services, 8(4), 229-255.
10. Paek, J., Kim, J., & Govindan, R. (2010, June). Energy-efficient rate-adaptive GPS-based positioning for smartphones. In Proceedings of the 8th international conference on Mobile systems, applications, and services (pp. 299-314). ACM.
11. Liu, J., Priyantha, B., Hart, T., Ramos, H. S., Loureiro, A. A., & Wang, Q. (2012, November). Energy efficient GPS sensing with cloud offloading. In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (pp. 85-98). ACM.
12. Ma, H., Zhao, D., & Yuan, P. (2014). Opportunities in mobile crowd sensing. IEEE Communications Magazine, 52(8), 29-35.
13. Wang, L., Zhang, D., Yan, Z., Xiong, H., & Xie, B. (2015). effSense: a novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading. IEEE Transactions on Systems, Man, and Cybernetics: Systems,45(12), 1549-1563.
14. Lane, N. D., Chon, Y., Zhou, L., Zhang, Y., Li, F., Kim, D., Cha, H. (2013, November). Piggyback CrowdSensing (PCS): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (p. 7). ACM
15. Wang, L., Zhang, D., Wang, Y., Chen, C., Han, X., & M'hamed, A. (2016). Sparse mobile crowdsensing: challenges and opportunities. IEEE Communications Magazine, 54(7), 161-167.

Author's Bio

Jiangtao Wang received his Ph.D. degree in Peking University, Beijing, China, in 2015. He is currently an assistant professor in Institute of Software, School of Electronics Engineering and Computer Science, Peking University. His research interest includes crowd sensing, ubiquitous computing, and cooperative computing.

Yasha Wang received his Ph.D. degree in Northeastern University, Shenyang, China, in 2003. He is a professor and associate director of National Research & Engineering Center of Software Engineering in Peking University, China. His research interest includes urban data analytics, ubiquitous computing, software reuse, and online software development environment. He has published more than 50 papers in prestigious conferences and journals, such as ICWS, UbiComp, ICSP and etc. As a technical leader and manager, he has accomplished several key national projects on software engineering and smart cities. Cooperating with major smart-city solution providing companies, his research work has been adopted in more than 20 cities in China.

Daqing Zhang is a professor at Peking University, China, and Télécom SudParis, France. He obtained his Ph.D from the University of Rome "La Sapienza," Italy, in 1996. His research interests include context-aware computing, urban computing, mobile computing, and so on. He served as the General or Program Chair for more than 10 international conferences. He is an Associate Editor for ACM Transactions on Intelligent Systems and Technology, IEEE Transactions on Big Data, and others.

Abdelsalam (Sumi) Helal (F'15) received the Ph.D. degree in computer sciences from Purdue University, West Lafayette, IN, USA. He is currently a professor in the School of Computing and Communications, Lancaster University, Lancaster, LA1 4YW, UK. His research interests include pervasive and mobile computing, the Internet of Things, smart spaces, smart city concepts, smart health and technologies for successful aging, and persuasive computing. He is the Fellow of the IEEE.