International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

EFFICIENT NEAR-OPTIMAL TASK ALLOCATION FOR PIGGYBACK CROWDSENSING

S.NITHYA [1] ,Dr.C.DEEPA[2]

[1]M.Phil Research Scholar, Department of Computer science,
Sri Ramakrishna College of Arts and Science [Formerly S.N.R SONS College],
Coimbatore-641006, TamilNadu, (India)
[2]Associate Professor,
Department of Information Technology,
Sri Ramakrishna College of Arts and Science [Formerly S.N.R SONS College],
Coimbatore-641006, Tamil Nadu, (India)

ABSTRACT

Mobile crowd sensing had a novel spatial-temporal coverage metric, k-depth coverage, problems.. This metric considers both the fraction of subareas covered by sensor readings and the number of sensor readings collected in each covered subarea. Then ICrowd, a generic MCS task allocation framework operating with the energy-efficient Piggyback Crowd sensing task model, is proposed to optimize the MCS task allocation with different incentives and k-depth coverage objectives/ constraints. ICrowd first predicts the call and mobility of mobile users based on their historical records, then it selects a set of users in each sensing cycle for sensing task participation, so that the resulting solution achieves two dual optimal MCS data collection goals like near-maximal k-depth coverage without exceeding a given incentive budget and near-minimal incentive payments while meeting a predefined k-depth coverage goal. The Ad-hoc On-demand Distance Vector (AODV) routing protocol is a routing protocol used for dynamic wireless networks where nodes can enter and leave the network at will. To find a route to a particular destination node, the source node broadcasts a RREQ to its immediate neighbors. If one of these neighbors has a route to the destination, then it replies back with a RREP. Otherwise the neighbors in turn rebroadcast the request. This continues until the RREQ hits the final destination or a node with a route to the destination. At that point a chain of RREP messages is sent back and the original source node finally has a route to the destination.

.***Keywords:Mobile CrowdSensing (MCS), MCS task allocation, incentives.***

I.INTRODUCTION

Mobile Crowdsensing (MCS) [3] has become an efficient way to sense and collect environment data of urban area in real-time (e.g., air quality, temperature or noise level). Instead of deploying static and expensive sensor network in urban area, MCS leverages the sensors embedded in mobile phones and the mobility of mobile users to sense their surroundings, and utilizes the existing communication infrastructure (e.g., 4G, Wi-Fi etc.) to

collect data from mobile phones scattered in the urban area. By collecting sensor readings from mobile users, a "big picture" of the environment in the target area can be obtained using MCS without significant cost.

iCrowd—a near-optimal task allocation framework for mobile crowdsensing, which can improve the efficiency of environment data collection with less cost. Here we first discuss the motivations and background of our MCS research, then we formulate a new MCS research problem with a unified set of research assumptions and objectives. We elaborate the technical challenges of the proposed research and finally we summarize our technical contributions.

In MCS, there are two main players: MCS organizer who is the person or organization coordinating the sensing task, and MCS participants who are the mobile users involved in the sensing task. An MCS task usually requires the organizer to recruit participants, to allocate sensing tasks to selected participants, and to collect sensor readings from these participants' mobile devices that well represent the characteristics of the target sensing region [4], often with budget constraints on participant incentives.

Specifically, the MCS organizer needs to specify the target sensing area, which often consists of a set of subareas, and further specify the sensing duration (e.g., 10 days), which is usually divided into equal-length sensing cycles (e.g., each cycle lasts for an hour). Once the settings of subareas and sensing cycles are determined, the MCS application usually needs to collect a number of sensor readings from each subarea of the target region in each sensing cycle. Taking a one-week urban air quality monitoring MCS task as an example, the MCS organizer first divides the whole area into 1 km2 grid cells, then splits the oneweek MCS study time into a sequence of one-hour sensing cycles [5], and further requests at least one MCS participant in each grid to upload the air quality sensor reading in each sensing cycle. In this case, however, the cost of the whole MCS task, including the energy consumption caused by the MCS application on each participant's mobile device and the overall incentives cost to recruit participants, could be quite high. In order to lower the cost of MCS, the mechanism to reduce the energy consumption and control the overall incentive cost, while ensuring the spatial-temporal coverage of collected environment sensor readings, is thus needed. Next we introduce the background of our research from following thee aspects:

Energy-efficient piggyback crowdsensing (PCS). So far, various solutions have been proposed to reduce energy consumption of individual mobile device, ranging from adapting sensing frequency to inferring part of the data rather than sensing and uploading all data [6], [7]. One of the effective solutions is Piggyback Crowdsensing, which reduces energy consumption by leveraging smartphone opportunities to perform sensing tasks and return sensor readings [8],[9]. For example, uploading sensing data in parallel with a 3G call can reduce about 75 percent of energy consumption in data transfer compared to the 3G-based solution [9].

Spatial-temporal coverage of MCS tasks. The typical approach for measuring the spatial-temporal coverage is to use the fraction of subareas being covered by at least one sensor reading in each sensing cycle. AnMCS application may need to collect sensor readings to achieve either full spatial-temporal coverage [10], [11] or partial spatial-temporal coverage [12], [13], [14], [15]. Usually, the use of full spatial-temporal coverage is to ensure the collected sensor readings representing each subarea in each sensing cycle, while the use of partial coverage aims to collect data that could represent a certain fraction (e.g., 80 percent) of subareas in each cycle.

Incentives, budget and task allocation. In order to recruit participants for MCS, each selected participant is typically offered a certain amount of money as incentives and thus the MCS organizer needs to prepare a budget equal to the total incentives paid to all participants in each MCS task. Once the spatial-temporal coverage and total budget are determined, the MCS organizer needs to select participants with the goal minimizing the total budget while ensuring the spatial temporal coverage, or maximizing the spatial-temporal coverage with a fixed budget.

In order to achieve either of above goals, given users who are willing to participate the MCS task, an MCS organizer needs to allocate sensing tasks to users, where the organizer first selects participants for MCS [1], and then decides in which sensing cycles each participant should perform the MCS task [14]. Only the participants selected for MCS will be paid with incentives.

With all aforementioned coverage, energy, and incentives issues, we are motivated to study the problem of optimizing MCS tasks, subject to various spatial-temporal coverage and incentive cost objectives/constraints.

II.LITERTURE SURVEY

D. Zhang, H. Xiong, L. Wang, and G. Chen[1] proposes a novel participant selection framework, named CrowdRecruiter, for mobile crowdsensing. CrowdRecruiter operates on top of energy-efficient Piggyback Crowdsensing (PCS) task model and minimizes incentive payments by selecting a small number of participants while still satisfying probabilistic coverage constraint. In order to achieve the objective when piggybacking crowdsensing tasks with phone calls, CrowdRecruiter first predicts the call and coverage probability of each mobile user based on historical records. It then efficiently computes the joint coverage probability of multiple users as a combined set and selects the near-minimal set of participants, which meets coverage ratio requirement in each sensing cycle of the PCS task. We evaluated CrowdRecruiter extensively using a large-scale real world dataset and the results show that the proposed solution significantly outperforms three baseline algorithms by selecting 10.0% - 73.5% fewer participants on average under the same probabilistic coverage constraint.

H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier[2] described a novel task allocation framework, CrowdTasker, for mobile crowdsensing. CrowdTasker operates on top of energy-efficient Piggyback Crowdsensing (PCS) task model, and aims to maximize the coverage quality of the sensing task while satisfying the incentive budget constraint. In order to achieve this goal, CrowdTasker first predicts the call and mobility of mobile users based on their historical records. With a flexible incentive model and the prediction results, CrowdTasker then selects a set of users in each sensing cycle for PCS task participation, so that the resulting solution achieves nearmaximal coverage quality without exceeding incentive budget. We evaluated CrowdTasker extensively using a large-scale realworld dataset and the results show that CrowdTasker significantly outperformed three baseline approaches by achieving 3% - 60% higher coverage quality.

R.K. Ganti, F. Ye, and H. Lei[3] implemented an emerging category of devices at the edge of the Internet are consumer centric mobile sensing and computing devices, such as smartphones, music players, and in-vehicle sensors. These devices will fuel the evolution of the Internet of Things as they feed sensor data to the Internet at a societal scale. In this paper, we will examine a category of applications that we term *mobile crowdsensing*,

where individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest. We will present a brief overview of existing mobile crowdsensing applications, explain their unique characteristics, illustrate various research challenges and discuss possible solutions. Finally we argue the need for a unified architecture and envision the requirements it must satisfy.

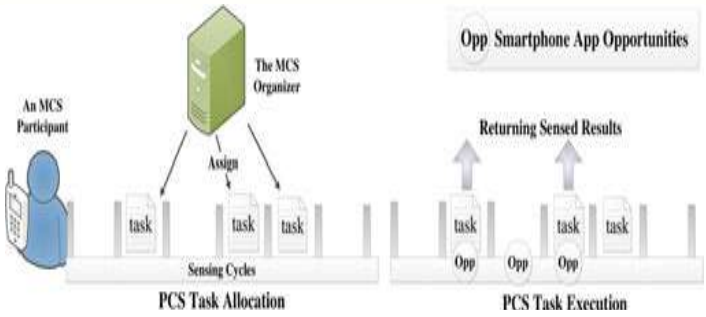
M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A.T Campbell[4] described Continuous sensing applications (e.g., mobile social networking applications) are appearing on new sensor-enabled mobile phones such as the Apple iPhone, Nokia and Android phones. These applications present significant challenges to the phone's operations given the phone's limited computational and energy resources and the need for applications to share real-time continuous sensed data with back-end servers. System designers have to deal with a trade-off between data accuracy (i.e., application fidelity) and energy constraints in the design of uploading strategies between phones and back-end servers. In this paper, we present the design, implementation and evaluation of several techniques to optimize the information uploading process for continuous sensing on mobile phones. We analyze the cases of continuous and intermittent connectivity imposed by low-duty cycle design considerations or poor wireless network coverage in order to drive down energy consumption and extend the lifetime of the phone. We also show how location prediction can be integrated into this forecasting framework. We present the implementation and the experimental evaluation of these uploading techniques based on measurements from the deployment of a continuous sensing application on 20 Nokia N95 phones used by 20 people for a period of 2 weeks. Our results show that we can make significant energy savings while limiting the impact on the application fidelity, making continuous sensing a viable application for mobile phones. For example, we show that it is possible to achieve an accuracy of 80% with respect to ground-truth data while saving 60% of the traffic sent over-the-air.

III PROPOSED WORK

iCrowd—a near-optimal task allocation framework for mobile crowdsensing, which can improve the efficiency of environment data collection with less cost. Here we first discuss the motivations and background of our MCS research, then we formulate a new MCS research problem with a unified set of research assumptions and objectives. We elaborate the technical challenges of the proposed research and finally we summarize our technical contributions. We propose to study a novel MCS task allocation problem for Piggyback Crowdsensing applications, where we first assume that each MCS participant senses and uploads sensor readings leveraging smartphone opportunities (e.g., placing a 3G call) to reduce the MCS energy consumption.

We can increase the data delivery ratio and reduce the effects of packet loss caused by the node mobility. Specifically, the framework considers the regularity in mobility patterns during the construction of the routing tree and deployment of nodes. It also includes an overhearing mechanism for mobile nodes to further improve the data delivery ratio.

International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354



3.1 OBJECTIVE

k -depth coverage of MCS tasks. While the existing spatial Temporal coverage metrics usually assume that the environment data (e.g., air quality) of a subarea in a sensing cycle could be represented by a single sensor reading, it is reasonable to believe that the each subarea could be better characterized if we could deduce the environment characteristics using multiple sensor readings collected from the same subarea. However, if we increase the number of sensor readings in a subarea above a certain threshold, the accuracy of the deduced value may not increase anymore [16]. Thus we propose a novel spatial-temporal coverage metrics–i.e., k-depth coverage, which could be used as either an objective

3.2 UTILITY-BASED USER-CYCLE COMBINATION SELECTION ALGORITHM

Xn–the set of user-cycle combinations already selected in the nth outer loop, U—the overall set of users, I—total number of sensing cycles, and bo—the incentive payment for bonus.

Output: X0—a new set of user-cycle combinations selected in the current inner loop

1 begin
/* initialize */
2 X0 ;;
/* getting all users in Xn */
3 Sn getAllUsers(Xn);
/* getting all possible user cycle-combinations */
4 C fhu; iij8u 2 U; 0 _ i < Ig;
5 if bo ¼ 0 then
/* when bo ¼ 0 (i.e., Fixed Individual Incentive Setting), select a new user
(with all cycles) having the maximal utility */
6 u0 argmaxu2UnSn P 0_j<I Utility(hu; jijXn);
7 X0 fhu0; jij0 _ j < Ig;
8 else
/* when bo > 0 (i.e., Varying Individual Incentive Setting), select a new usercycle
combination having the maximal utility */
9 hu0; i0i argmaxu2CnXn Utility(hu; iijXn);X0 fhu0; i0ig;
10 return X0;

In this way, the inner-loop greedy process continues selecting/ adding a new subset of user-cycle combinations and deciding whether new user-cycle combinations should be added using the constraint-based stopping criterion, until the corresponding constraint-based stopping criterion algorithm decides to stop selecting new combinations.

Convergence-based outer-loop stopping criterion –Given the selected set of user-cycle combinations (e.g., Xn in the nth outer-loop iteration), the algorithm decides whether to return the task allocation results or continue for further computation. When bo ¼ 0 – i.e., the incentive to each participant is fixed, the algorithm stops at the first outer-loop iteration and returns X1 directly as the task allocation result. When bo > 0 – i.e., the individual incentive is dependent on the number of participating cycles, the algorithm needs to decide if to return the task allocation result or continue to obtain Xnþ1, with respect to the two MCS data collection goals.

3.3 AODV ROUTING PROTOCOL

The Ad-hoc On-demand Distance Vector (AODV) routing protocol is a routing protocol used for dynamic wireless networks where nodes can enter and leave the network at will. To find a route to a particular destination node, the source node broadcasts a RREQ to its immediate neighbors. If one of these neighbors has a route to the destination, then it replies back with a RREP. Otherwise the neighbors in turn rebroadcast the request. This continues until the RREQ hits the final destination or a node with a route to the destination. At that point a chain of RREP messages is sent back and the original source node finally has a route to the destination.

We proved that AODV protocol never produces routing loops by proving that a combination of sequence numbers and hop counts is monotonic along a route. This means that there can't be any loop in the routing table. The proof was done completely automatically and our algorithm was able to generate all the predicates needed. The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it. As the RREP propagates back to the source, nodes set

up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hopcount, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

IV. RESULTS AND DISCUSSION

The randomly directed exploration protocol on the same simulation framework as the previous protocol. Since the randomly directed exploration protocol relies on a local network topology, the random graph model cannot server for its simulations. Instead, The take the unit-disk graph as the sole network scenario. The choose a constant node degree and select as the priority range of the protocol. As a result, there are an average 2.5 neighbors in the priority zone of a node.

4.1 BASELINES AND COMPARISONS FOR GOAL. 2

In order to evaluate performance of iCrowd for Goal. 2, we build baselines and compare their performance to iCrowd, using following k-depth coverage constraint (k; r) and incentives (ba; bo) settings:

The base and bonus incentives are fixed to ba ¼ 1 and bo ¼ 0 respectively;

The k-depth coverage constraint ðk; rÞ is set to coverage depth k ¼ 1 and coverage ratio r ¼ 85% and 95%.

In this case, the incentive payment to each participant is fixed at 1 and the overall incentive payment is equal to the number of selected participants. In terms of coverage, the depth is set to k ¼ 1, thus the MCS task is targeted at collecting at least one sensor reading from at least a predefined percentage (r ¼ 85% or 95%) of subareas in each sensing cycle. With these incentive and k-depth coverage settings, the objective of this evaluation is to select a minimal number of participants and allocate tasks to each sensing cycle of each selected participant, while ensuring the selected participants covering at least 85 or 95 percent subareas in each sensing cycle. We first introduce the baseline algorithms for Goal. 2 that can select a minimal number of participants while ensuring a predefined percentage (e.g., r ¼ 95% or 85%) of subareas being covered by at least one sensing cycle (i.e., k ¼ 1) in each sensing cycle, then we compare the overall incentive payment consumed by baselines and iCrowd under the same coverage setting.

Baselines for Goal. 2 – In our evaluation, we provide three baseline methods with different utility-based selection strategy from iCrowd, but all of them share the same iteration process and stopping criterion.

1) MaxMin – MaxMin uses a single-loop greedy algorithm to select an unselected user and all his/her cycles in each iteration, until the k-depth coverage constraint achieved. Specifically, the algorithm, selects/adds a user having the maximal utility min0_i<IP(X [fhu; iig; k; r; i) in each iteration. Note that the utility function used

here refers to the minimum of the coverage probabilities among all sensing cycles. Please refer to [29] for details.

2) MaxCom – The basic idea of MaxCom is to select the next participant who best complements with the selected set of participants in terms of coverage probability. This method is derived from the idea proposed by [21].

3) MaxCov – In each MaxCov is to simply select the next participant who covered the most cell towers in the historical call traces, among all the unselected mobile users.

Performance Comparisons for Goal. 2: Overall Incentive Payment Comparisons under the same k-depth coverage constraint - In Table 3 , we present the performance comparison on overall incentives consumption (i.e., number of selected participants for each of the four tasks) between iCrowd and baselines. It is clear that iCrowd outperformed MaxMin, MaxCom and MaxCov methods in all PCS tasks. On average, iCrowd consumed 10.0-21.5 percent less overall incentives compared to MaxMin (i.e., 10.0-21.5 percent fewer selected participants), consumed 23.7-43.5 percent less overall incentives compared to MaxCom, and consumed 54.2-73.5 percent less overall incentives compared to MaxCov. In terms of k-depth coverage, we show the Max/Min/Average the percentage of cell towers covered by at least one sensor reading (i.e., k ¼ 1) in each sensing cycle in Fig. 5 using iCrowd and baselines. For all sensing cycles, the required coverage ratio (i.e., 95 and 85 percent) are achieved by the four methods without significant differences.

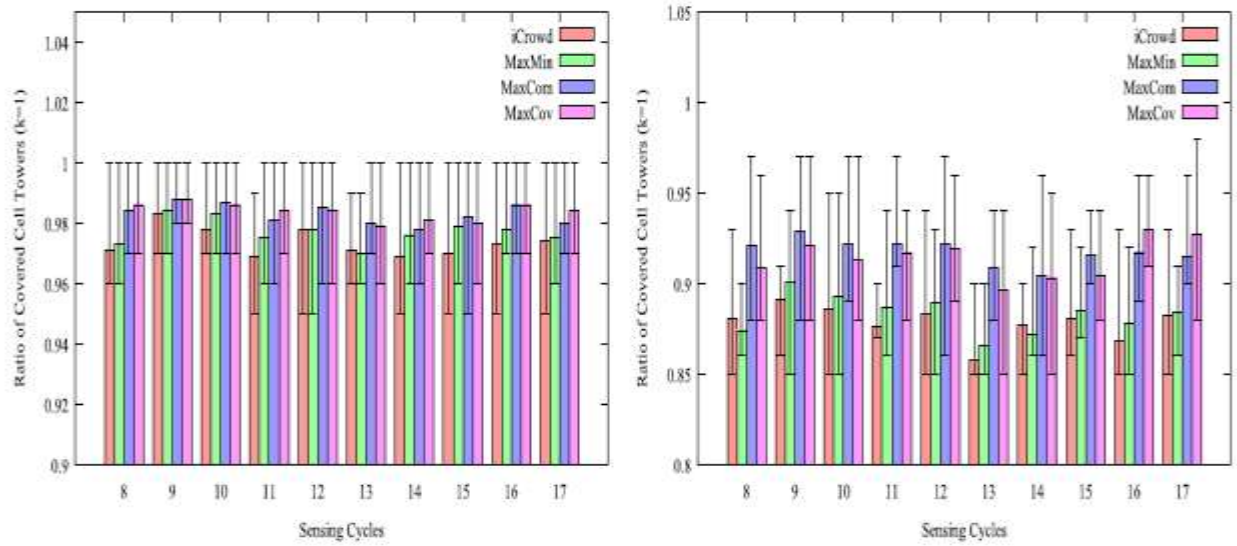


Fig. 5. Max/Min/Average ratio of covered cell towers based on the MERGED region (Please see the result of RESIDENTIAL and BUSINESS regions in Appendix, available in the online supplemental material).

4. 2 TEMPORAL COVERAGE OF SENSOR READINGS

As the coverage ratio r specified in this paper is less than 100 percent, it is conceivable that some cell towers may have low temporal coverage or zero coverage (e.g., no sensor readings in any sensing cycle). Thus we examined the temporal coverage of the cell towers using the three datasets when Ratio = 85%. As shown in Fig.

6, most cell towers can be covered in more than 80 percent sensing cycles when using the BUSINESS, RESIDENTIAL and MERGED datasets. The two least covered cell towers (tower id = 724 and id = 646 in the MERGED region) were still covered in 59 percent of the sensing cycles.

Due to space limitation, some iCrowd evaluation results are not reported here. Readers are encouraged to see the Appendix, available in the online supplemental material, for additional details including the time consumption of iCrowd for both Goal. 1 and Goal. 2 and other experiment insights.

(a) BUSINESS 85% (b) RESIDENTIAL 85% (c) MERGED 85%

Fig. 6. Temporal coverage ratio of cell towers in BUSINESS, RESIDENTIAL, and MERGED regions.

V.CONCLUSION AND FUTUREWORK

The unified task allocation framework, iCrowd, for Piggyback Crowdsensing. iCrowd is designed to optimally allocate sensing tasks to PCS participants, subject to different incentive and spatial-temporal coverage constraints/objectives. Specifically, iCrowd could be adopted to either maximize the overall k-depth coverage across all sensing cycles with a fixed budget or to minimize the overall incentive payment while ensuring a predefined k-depth coverage constraint, by selecting a number of participants and determining in which sensing cycles each selected participant is needed for the PCS task participation. The PCS was adopted to reduce energy consumption of individual mobile device, by exploiting call opportunities to perform sensing tasks and upload sensed data. In order to allocate PCS task for either optimal MCS data collection goals, iCrowd first predicts the coverage probability of each mobile user, then performs a near-optimal participant/cycle task allocation search algorithm with low computational complexity. Theoretical analysis proves that iCrowd can achieve near-optimality for both optimal MCS data collection goals, and evaluations with a large-scale real-world dataset show that iCrowd outperformed six baseline approaches. For Goal. 1 it achieved 3-60 percent higher k-depth coverage compared to baseline approaches under the same budget constraint, while for Goal. 2 iCrowd required 10.0-90.5 percent less overall incentive compared to baselines under the same k-depth coverage constraint.

In future the encryption and decryption process will be developed. This will avoid the node failures and collision occurred within the network. This encryption process secures the data when it is transmitting through the network. Also the future system will be doing the efficient algorithm for the advanced encryption process.

REFERENCES

[1] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., 2014, pp. 703–714.

[2] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, "Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint," in Proc. IEEE Int. Conf. Pervasive Comput. Commun., 2015, pp. 55–62

[3] R.K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," IEEE Commun. Mag., vol. 49, no. 11, pp. 32–39, Nov. 2011.

[4] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A.T Campbell, "Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones," Pervasive Comput., 2010, pp. 355–372.

[5] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 1436–1444.

[6] S. Reddy, D. Estrin, and M. Srivastava,, "Recruitment framework for participatory sensing data collections," in Proc. 8th Int. Conf. Pervasive, 2010, pp. pp. 138–155

[7] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, "Fostering participaction in smart cities: A geo-social crowdsensing platform," IEEE Commun. Mag., vol. 51, no. 6, pp. 112–119, Jun. 2013

[8] X. sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in Proc. IEEE Conf. Comput. Commun., 2012, pp. 1916–1924.

[9] D. Philipp, J. Stachowiak, P. Alt, F. D€urr, and K. Rothermel, "Drops: Model-driven optimization for public sensing systems," in Proc. IEEE Int. Conf. Pervasive Comput. Commun., 2013, vol. 18, p. 22.

[10] A. Singla and A. Krause, "Incentives for privacy tradeoff in community sensing," in Proc. 1st AAAI Conf. Human Comput. Crowdsourcing, 2013, pp. 165–173.