Short Paper: Dynamic QoS-Aware Event Sampling for Community-based Participatory Sensing Systems*

Ioannis Boutsis
Department of Informatics
Athens University of Economics and Business
Athens, Greece
mpoutsis@aueb.gr

Vana Kalogeraki
Department of Informatics
Athens University of Economics and Business
Athens, Greece
vana@aueb.gr

ABSTRACT

Over the recent years, the proliferation of mobile networking and the increasing capabilities of smartphone devices have led to the development of the "Community-based Participatory Sensing" approach, where users participate in data collection and sharing in a wide range of application areas such as entertainment, transportation and environmental monitoring. This paper develops a participatory sensing system that uses a sampling mechanism that aims to stimulate user participation in dynamic groups that provide services and get compensated for the services they provide. Users participate in the community by sensing and sharing streams of events. The system then uses a sampling mechanism to define a subset of events that preserves the characteristics of the stream data and provides the highest "information gain" to the system, given the budget and resource constraints. Our experimental results illustrate that our approach is practical, efficient and depicts good performance.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]

Keywords

Distributed Systems, Mobile Systems, Community-based Participatory Sensing, Sampling, QoS

1. INTRODUCTION

In recent years, the proliferation of mobile networking and the increasing capabilities of smartphone devices have led to the development of a new class of "community-based participatory sensing" systems, where all members of the community contribute data to the system for the interest of the community. We have already witnessed this trend in a number of application domains that include location-based services such as personalized news feed and for identifying areas of good WiFi connectivity [3], earthquake warning detection systems [7], transportation systems [4] and gaming [9].

Encouraging individuals to participate in community-based data sensing and collection has important advantages in terms of improving the quality of the systems with user feedback and achieving rapid detection of events. We envision to make use of the rich suite of sensors on the smartphones, to extract application events of interest. However, we argue that each data stream has a different importance that varies relative to its content. The "real value" of the user data streams depends on several factors that include the data characteristics, the user context (*e.g.,* geographical position), available resources (*e.g.,* communication, energy), etc. Thus, a fundamental issue is how to choose a representative subset from the produced data streams.

Sensing and sharing data from mobile phones presents several challenges. Mobile devices often produce more data than the network can deliver or the application can process. From the user perspective that sense, collect and share their data, the benefit of the user is dependent not only on the compensation received, but also on the effort given by the user to collect such data. This includes resource costs, battery consumption or privacy concerns. The amount of data that an application is capable of processing is constrained by two major factors: First, the resource availability across the distributed system that will collect and process the stream data. Second, the cost for the application to receive the data. Hence, the system has to consider the amount of data streams that can be supported, so as to select those data streams that provide the highest information profit.

In this paper we present P-SenSe, a community-based participatory sensing system that aims to stimulate user participation by encouraging them to be members of the system as part of a dynamic group that provides services and get compensated for the services they provide. Users participate in the community by sensing and sharing streams of events. Users define their offers and they receive a monetary reward for the stream data (events) they provide. The offers reflect the quality of the data provided along with the corresponding cost for providing this data. The system then selects a subset of events that preserves the characteristics of the stream data and provides the highest "information gain" to the system, given the budget and resource constraints. Our experimental evaluation on PlanetLab illustrate that our approach is practical, efficient and depicts good performance.

*This research has been supported by the European Union through the Marie-Curie RTD (IRG-231038) project, a Hellenic Republic Ministry of Education, Lifelong Learning and Religious Affairs Thalis DISFER project and by AUEB through a PEVE project.

DEBS '12, July 16–20, 2012, Berlin, Germany.

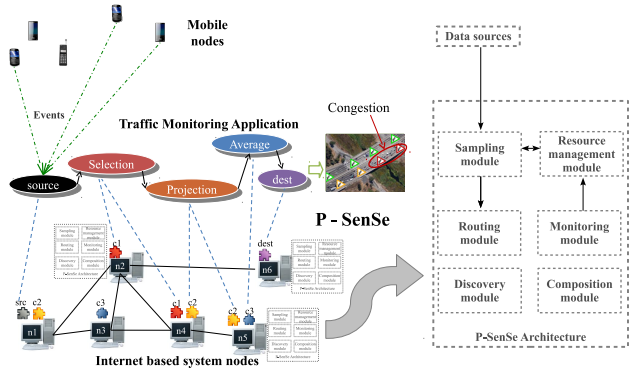

Figure 1: Our system architecture.

2. SYSTEM ARCHITECTURE AND MODEL

We have developed P-SenSe, a community-based participatory sensing system (shown in Figure 1) that comprise: i) A *discovery module* for discovering application components and data streams, ii) a *routing module* for routing data streams and protocol messages, iii) a *monitoring module* for maintaining resource utilization profiles, iv) a *composition module* that selects and instantiates application components, v) a *resource management module*, responsible for rate allocation based on the applications' QoS demands and resource availability, and vi) a *sampling module*, that determines which of the available data streams will be processed. The sampling is conducted in cooperation with the resource management module.

Our work focuses on streams of events generated by application modules running on the smartphones. A stream of events consists of a sequence of individual chunks of data, referred as *Application Data Units* (ADUs); the size of a data unit depends on the type of the application. Each event represents a short message that is triggered locally at the phone using sensing devices present on mobile phones such as microphone, camera, GPS, accelerometer and motion sensors. Examples of events are: <Video data, timestamp, latitude, longitude> (for surveillance monitoring applications).

Each user i defines an $offer_i(t)$ as a pair of values: (a) $QoI_i(t)$, and (b) $cost_i(t)$, for the produced $ADU_i(t)$ at time t. The offer reflects the user's willingness to share his sensed data with the community and depicts the importance of the specific stream data, along with the cost to the system to receive that stream. The $QoI_i(t)$ value is an application-defined functions, computed as defined in the next section. The $cost_i(t)$ can be expressed as a monetary payment that the user wishes to receive for providing the data stream. In our system we assume that all users are cooperative and that there are no malicious users. Thus, users submit their "true" $QoI_i(t)$ information and get compensated for providing that information.

The goal of the sampling scheme is to select the most appropriate subset of events s_i, (where $s_i < m_i$) for each region R_j. $Deadline_q$ is a relative metric that represents the end-to-end time constraint required for the application q to process a number of ADUs. The sampling component determines the maximum amount of events that the system can efficiently process, depending on application's q requested $Rate_q$, $Deadline_q$ constraints, and the system $Budget$.

3. INFORMATION QUALITY

The utility of the event to the system is concerned with the significance that the event has for the application. We assume that each stream of events has a "value" expressed as a function of time. Two fundamental questions appear when defining the utility of the information. First, not all stream data have the same utility (*i.e.,* importance) in the system. Typically, some data streams may have higher utility than others and this relative utility may change at runtime. Second, the relative utility of the stream data might not be directly related to the time deadline, within which the stream data needs to be delivered to the system. Our goal is to consider these two parameters of information utility and application time deadline in concert, when determining the appropriate events to be sampled that would result in providing the highest information profit to the system.

Utility functions. We use utility functions (that we call *QoI functions*) to express the benefit of providing a stream of events to the application. Although our proposed QoI functions are mainly linear functions over time, in the general form they might have different forms and shapes to be able to meet the demands of each application [8].

QoI Function Form. We employ a number of QoI functions, that, in their general form are expressed as:

$$QoI_i(t) = QoI_i(t-1) + \sum CurrentValues(t) \qquad (1)$$

In every run of the sampling mechanism the $QoI_i(t)$ value for a mobile node will be adjusted based on the Current Values of the defined functions that represent application specific utilities. When the mobile node is selected from the sampling component, its $QoI_i(t-1)$ is set to zero. In our application scenarios, we define the following QoI functions:

QoI Density function. The first QoI function that we define represents the density of the mobile phones in a given geographical region. Our aim is to avoid receiving stream data from all nodes in areas that are fairly dense. Thus, this function considers the number of mobile phones that provide information from the same location ($Neighbors_i$) in region R_j. We define the QoI equation:

$$QoI_i(t) = QoI_i(t-1) + (\#MaxNeighbors - \#Neighbors_i) \qquad (2)$$

In every period the $QoI_i(t)$ is adjusted based on the actual number and the maximum expected neighbors in the region. The number of neighbors for every mobile node can be either estimated using historical information about the region or can be retrieved from regional servers.

The amount of neighbors in a given region, where a mobile node is located, is expected to change dynamically due to the mobility of the users. Thus, the value of ($\#MaxNeighbors - \#Neighbors_i$) should change over time as the number of mobile nodes in a region changes dynamically, affecting the density of the region. Although the function increases linearly over time, it should have small fluctuations.

QoI Region function. This is the case where the system assigns different weights per region R_j according to the relative importance of the region ($RegionImportance_j$). This weight is typically predefined so that the mobile nodes can obtain it from the system (*e.g.,* in city areas the weight may be higher). If the weight needs to change at run-time it can be retrieved using regional servers. This function can be

expressed as follows:

$$QoI_i(t) = QoI_i(t-1) + RegionImportance_j \quad (3)$$

The above utility function considers the weight of the corresponding regions. When the mobile phone i changes regions, the weight of the new region j is used and there is a new incremental step based on that weight. Although, the form of that utility function resembles the form of the density QoI function, the variance of the slope should be different, as the mobile node is expected to stay in the same region for several time units.

QoI Transition function. Our experimental evaluation indicated that the total system knowledge improves when the utility function also considers whether a mobile i has transitioned from one region to another. Hence, we define:

$$QoI_i(t) = QoI_i(t-1) + Transition_i \quad (4)$$

When a mobile node i changes region, the new values should be more important, since its last processed ADU is located in the previous region, which is no longer valid and the new region has no evidence of the mobile node and its data. Hence, both regions have erroneous data for the current moment. This function results in an improvement for the sampled results, when it is combined with other QoI functions.

4. SAMPLING

The goal of the sampling process is to select the events with the highest information quality, with respect to the system's budget and resource constraints. Let us consider a geographic area consisting of $R_1, R_2, .., R_j$ regions with M mobile nodes. Each of the mobile nodes i produces streams of events (ADUs) over time t with an information quality $QoI_i(t)$. At each time unit t the user defines his $offer_i(t)$ as a pair of the $QoI_i(t)$ value, that reflects the relative utility (importance) of the stream data, and the respective $cost_i(t)$ that depicts the profit the user estimates for the produced data unit. The value of the $QoI_i(t)$ metric is adjusted at different runs of the sampling mechanism. The higher the $QoI_i(t)$ value, the higher the chances of a mobile node to be selected for sampling. The cost could be either a fixed price payment provided by the system, provider-defined or user-defined.

The system needs to select the most appropriate events to maximize the information quality provided to the system. The selection of the events should also consider the resource constraints imposed by the system on the $Rate_q$ that the system can support so that the application q can meet its relative $Deadline_q$, determined by the resource management module. Thus, we aim to select these ADUs that maximize the sum of the utilities that each of the mobile nodes i can provide at time t ($\sum_{i \in M} QoI_i(t)$), for the selected ADUs, so that their cost won't exceed the Budget ($\sum_{i \in M} cost_i(t) \leq Budget$) and their rate will not exceed the rate that the system can support ($\sum_{i \in M} ADU_i(t)/Deadline_q \leq Rate_q$).

All nodes that wish to participate in the sampling transmit their offers to the sampling components. The resource management module informs the sampling components about the rate $Rate_q$ that the system is able to support to meet $Deadline_q$. Thus, the sampling component will be able to select the nodes with the highest offers, considering the corresponding system resource constraints. Once selected, the nodes will send their $ADU_i(t)$, get paid based on the corresponding $cost_i(t)$ and reset their $QoI_i(t-1)$ value to zero. Nodes are allowed to participate in the next round of sampling even though they were selected in the current round.

5. EXPERIMENTAL EVALUATION

Sampling Error Definition. We define the sampling error metric Δ_i as the expected absolute difference between the estimated average (y_i') and the exact average (y_i) of the records: ${\Delta_i}^2 = E[(y_i' - y_i)^2]$. The estimated average value of all the m_i data streams generated in a specific time period in region R_j, is computed by sampling $\bar{s_i}$ ($\bar{s_i} < m_i$) data streams$(x_1', x_2', .., x_{\bar{s_i}}')$, as $y_i'(y_i' = \sum_{k=1}^{\bar{s_i}} x_k'/\bar{s_i})$.

Experimental Setup: We have implemented our techniques over the P-SenSe middleware and tested it on the PlanetLab testbed. For the experiments we used the Berkeley's Mobile Millennium Dataset [5], a real time traffic data taken from GPS-enabled phones. The application scenario was implemented with 4 main components, shown in Figure 1: a *selection component*, a *projection component*, an *average component*, while the *dest component* receives this data to extract the traffic result map and define congested areas. Each experiment was run 5 times and the results presented are the average over all runs.

Experimental Results: We evaluate the behavior of several QoI functions under the same resource conditions, where we use a sample size of approximately 25%, and we set the same price for all ADUs, so the evaluation can be independent of the ADU cost.

We present 4 QoI functions: (1) **Random QoI** where the selection is made randomly, based on resource and budget constraints but without using information criteria, (2) **Region QoI** where each mobile receives a specific weight over time based on the region it belongs, whose value increases in accordance to the Region number. (3) **Region+Density QoI** where both region and density weights are computed for the mobile node and (4) **Region+Density+Transition QoI** where the 3rd QoI function is extended with a weight whenever a mobile node changes region. Here, we aim at reducing the Sampling Error in each region with respect to the region's importance. We denote that the importance increases along with the region's name. For example, Region 5 is more important in our application than Region 2.

Figures 2, 3, 4 and 5 illustrate the Average Error in km/h for each region over all runs. Figure 2 presents the behavior of the Region QoI function, where we observe an increase in the accuracy for specific regions, due to the nature of the QoI function. However, this accuracy improvement clearly depends on the specific regions, as there can be regions with fewer ADUs sampled or with varying speeds of mobile nodes. Moreover, combining the density weight along with the region one in Figure 3 depicts an improvement in the regions with lower weights. This derives from the fact that although we still provide increased accuracy in the higher weighted regions, when the instant density is low in one region, its QoI value increases faster.

Figure 4 shows that when the Transition weight was integrated it resulted to a significant accuracy improvement. This derives from the fact that when a mobile node changes region, there is an error deviation not only to its previous region, since its last transmitted ADU should not be taken into account, but also to the new region that has no evidence for that node.

Finally, Figure 5 presents the total average error using the random QoI. In contrast to other QoIs, where the errors re-

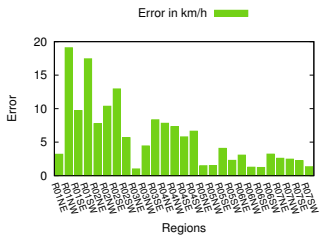

Figure 2: Region QoI

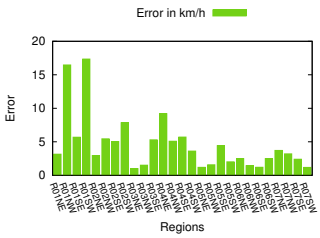

Figure 3: Region + Density QoI

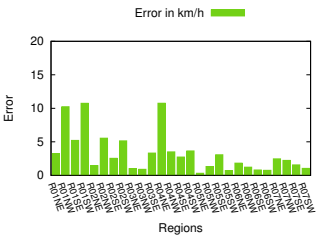

Figure 4: Region + Density + Transition QoI

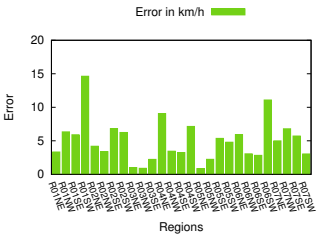

Figure 5: Random QoI

main almost identical for the same experiments, the random QoI has a different output in every run. Thus, the weights on every region vary significantly in the experiments. We should note, that our goal was to provide a level of accuracy, especially in high weighted regions. As can be observed, we have achieved to approach the application logic in all of our QoI formulations, in contrast to the random QoI, which cannot guarantee such a behavior.

6. RELATED WORK

Participatory Sensing systems have recently become extremely popular for processing high-throughput, low-latency data streams and a number of systems have emerged in the literature [4], [10].

The research in the area involving the problem of Sampling is very rich and several approaches have been proposed. In [6], the authors perform region sampling in sensor networks, in order to reduce the energy cost rate and use statistics to predict the optimal sampling plan. Our approach outperforms that since it is able to use different types of QoI functions to implement the sampling. Moreover, their goal is to bound the energy consumption while minimizing the approximation error, while our goal is to maximize the information benefit of the system. Al-Kateb *et al* in [1] propose an algorithm to extend the reservoir sampling, that selects a uniform random sample of a given size from an input stream of an unknown size, with an adaptive-size reservoir. However, our technique driven by the application logic and using the information quality, outperforms uniform random samples. Arai *et al* in [2] propose a technique for sampling in aggregation queries. Their approach suggests a Peer-To-Peer database as an infrastructure which is opposed to the stream processing architectural logic. Another sampling technique is to use a statistical model to capture the correlations between the different sensors readings. Finally, Wu *et al* in [11] suggest an approach where the Quality of Data of queries compared to the processing cost is taken into consideration. However, they focus on the quality profit of each query rather than the profit of the entire system.

7. CONCLUSIONS

In this paper, we have presented P-SenSe, a system that aims to improve user participation in community-based participatory sensing systems. P-SenSe makes it easy for users to sense, collect and share streams of data and get compensated for the data they provide. We propose algorithms that determine a suitable sample of the events based on the information quality that the individual nodes provide in the system and the corresponding costs. We have observed experimentally that the QoI function that combines density, region and transition, resulted in higher accuracy. As expected, the Random QoI function results in the worst behavior because it does not consider neither the application logic nor the importance of the events when sampling.

8. REFERENCES

[1] M. Al-Kateb, B. S. Lee, and X. S. Wang. Adaptive-size reservoir sampling over data streams. In *SSDBM*, page 22, Banff, Canada, July 2007.

[2] B. Arai, G. Das, D. Gunopulos, and V. Kalogeraki. Approximating aggregation queries in peer-to-peer networks. In *ICDE*, page 42, Atlanta, GA, USA, April 2006. IEEE Computer Society.

[3] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikinen, V. Tuulos, S. Foley, and C. Yu. Data clustering on a network of mobile smartphones. In *SAINT*, Munich, Germany, July 2011.

[4] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G. seop Ahn, and A. T. Campbell. Metrosense project: People-centric sensing at scale. In *SenSys*, Boulder, Colorado, USA, Oct-Nov 2006.

[5] J. Herrera. Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment. In *Transport. Res. Part C*, 2009.

[6] S. Lin, B. Arai, D. Gunopulos, and G. Das. Region sampling: Continuous adaptive sampling on sensor networks. In *ICDE*, Cancún, México, April 2008.

[7] M. Olson, A. H. Liu, M. Faulkner, and K. M. Chandy. Rapid detection of rare geospatial events: earthquake warning applications. In *DEBS*, pages 89–100, New York, USA, July 2011.

[8] B. Ravindran, E. D. Jensen, and P. Li. On recent advances in time/utility function real-time scheduling and resource management. In *ISORC*, pages 55–60, Seattle, WA, USA, May 2005.

[9] A. Schmieg, M. Stieler, S. Jeckel, P. Kabus, B. Kemme, and A. P. Buchmann. psense - maintaining a dynamic localized peer-to-peer structure for position based multicast in games. In *Peer-to-Peer Computing*, pages 247–256, Aachen, Germany, September 2008.

[10] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys*, Berkeley, California, USA, November 2009.

[11] H. Wu, Q. Luo, J. Li, and A. Labrinidis. Quality aware query scheduling in wireless sensor networks. In *DMSN*, Lyon, France, August 2009.