Did Anybody See That? Smartphone Tracking for Historical Data Retrieval *

Vikram P. Munishwar
Computer Science
State University of New York at Binghamton
vmunish1@cs.binghamton.edu

Nael B. Abu-Ghazaleh†
School of Computer Science
Carnegie Mellon University, Qatar
naelag@cmu.edu

ABSTRACT

Smartphones have revolutionized the way in which sensing has been performed traditionally. The people-centric nature of smartphone-based sensing enables them to be a part of participatory or opportunistic sensing, where data is collected on a set of designated smartphones and delivered to a server. In this work, we identify the existence of another type of behavior, where the data is not delivered but archived locally on the phones for later retrieval. This type of behavior is common when the phone users capture some data (e.g. a video clip) out of their own interest. However, this complicates the future data retrievals due to the uncontrolled mobility of the data-capturing smartphones. Specifically, the research challenges for later data retrieval include finding the current locations of the required subset of the mobile phones that were present in a specific region at a specific time, without compromising location and identity privacy of the phone user. We discuss existing as well as novel architectural alternatives that can be used to address this problem, along with their qualitative evaluation.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*distributed network, centralized network*; H.3.3 [**Information Storage And Retrieval**]: Information Search And Retrieval—*search process, selection process*

General Terms

Design, Security

Keywords

Smartphone, Indexing, Sensor Network, Network Architecture

1. INTRODUCTION

Recent advances in the processing, communication, and sensing resources on smart phones coupled with their ubiquity and ease-of-use have added new dimensions to the traditional sensing mechanisms [1–4]. For the traditional sensing mechanisms, the focus is on deploying and tasking static or mobile sensors specifically to fulfill the underlying application's requirements. However, with the advent of human-carried devices, such as smartphones, equipped with a number of sensors, such as camera, microphone, accelerometer, magnetometer, gyroscope, etc., the notion of sensing has evolved to introduce many new sensing opportunities. Some of the applications of smartphone based sensing include understanding information about the user's context [14], understanding interactions between people and surroundings [4], and participating in the potentially large-scale active sensing operations [3].

The new sensing mechanisms can be categorized into participatory and opportunistic sensing, depending on the extent of peoples' participation in the sensing activity [6]. Participatory sensing generally involves selecting a group of people to actively participate in sensing useful data for an application [3]. On the other hand, in opportunistic sensing, the person carrying a smartphone does not need to actively participate in the sensing act, but the device itself activates itself at the time of appropriate sensing opportunities defined by an application [5]. While both the approaches differ in their sensing mechanisms, the subsequent data management part is still the same – to deliver the sensed data in real-time or in delay-tolerant fashion to the intended recipient(s).

Although, the real-time data reporting is necessary in a number of applications [7, 10], where the real-time updates can be used as alerts or as feedback mechanisms for actuators, in many occasions the data needs to be stored locally for possible later retrieval. Human-carried phones may store the data locally because: (1) the data is captured out of the phone user's interest, and not because of any underlying task; (2) the importance of the captured data is unknown, since, for instance, the data may be redundant; or (3) the size of the captured data is simply too large to be able to send it, given the energy and bandwidth limitations of the smartphone.

In this work, we focus on the cases where data (or event) is captured by people and stored locally on the smartphone. In such scenarios, it is necessary to send queries to the desired (or target) phones who could have been present at a location of interest at a given time, in order to obtain more information about the event that happened at that location and time. Examples of such location-time specific queries – spatio-temporal queries – include: *Is there a video footage available that was captured immediately before or at the time of an accident, in order to help further investigations?* or simply *How many people were present for yesterday's fire-*

*This research work was funded by Qatar National Research Fund (QNRF) under the National Priorities Research Program (NPRP) Grant No.:08-562-1-095

†This work was conducted while Dr. Abu-Ghazaleh was on a leave of absense from the State University of New York at Binghamton.

works show? As it can be noted, spatio-temporal queries may involve querying historical data, and thus, it is necessary to understand the current locations of the phones that were present at the event site during the event time.

While, spatio-temporal queries over target/event mobility are supported for the traditional sensor networks [12, 16], they present a significant challenge for people-centric sensor networks due to the uncontrolled mobility of the phone users. Specifically, the challenges include obtaining the set of people who were present in the given spatio-temporal window (at the given location and the time) of a historic or an old event, and deducing their current locations in order to be able to send queries to them. Furthermore, not everyone who was present at the location of the event would have captured the required event data. Thus, it is necessary to reduce the search space by ignoring the mobile phones who may have not captured any useful information. This can enable choosing the k most useful mobile phones for resolving the query. Finally, the data collection operation should maintain the location and identity privacy of the target smartphone user. We discuss these challenges in detail in Section 3.

We present a solution space in terms of the architectural alternatives to address above challenges for efficiently locating the target mobile phones. Specifically, we present and qualitatively evaluate existing architectures as well as a novel semi-centralized architecture, *MobiTrail*, that is based on storing the indexing structure at the intermediate access points. Details of the existing architectures are presented in Section 2, while the solution space for architectural alternatives is presented in Section 4. Finally, concluding remarks and directions for future research are given in Section 5.

2. RELATED WORK

In this section, we present an overview of architectural aspects of existing works related to smartphone networks.

Opportunistic sensing is a form of people-centric sensing, where human-carried mobile phones are tasked to perform sensing activities opportunistically, whenever the application-driven context requirements are fulfilled [4,5]. MetroSense [7] proposes a three-tier architecture for people-centric sensor networks of mobile phones. The lowest-tier consists of mobile phones and static sensors already deployed in the region of interest. The middle-tier comprises of Sensor Access Points (SAPs) deployed separately or integrated with the existing communication infrastructure such as WiFi access points. The roles of a SAP include sensing, tasking or programming sensors, collecting data from them, and providing secure and trusted interactions with them. The upper-tier is a server-tier, where one or multiple Ethernet-connected resource-rich servers act as a core component of the system by providing administrative functionalities. BikeNet [8] also uses opportunistically encountered WiFi access points for delay-tolerant data communication, while the commonly available cellular data channel for real-time data reporting.

Participatory sensing [3] enables creating a community-oriented sensor network that can gather, analyze, and share the local knowledge. It uses pre-deployed WiFi access points or cellular network for data collection purposes. Another form of participatory sensing, Micro-Blog [10], enables smart phone users to upload the sensory data along with location and time information. This data can be geo-tagged to enable viewing the world at a higher resolution. Microblogs are nothing but the user blogs enriched with sensory inputs such as multimedia data associated with location and time. Such microblogs are added to a central database system via the existing WiFi or cellular networks.

MobiSoC [11] presents a middleware that enables mobile social computing applications to help people reconnect with their physical communities and surroundings by leveraging information about peoples' locations, and their social relationships. MobiSoC also presents a centralized architecture, where people can communicate with MobiSoC via the existing WiFi network.

Essentially, almost all of the existing architectures use a centralized approach for data collection, where the sensed data is uploaded to a central unit periodically or opportunistically from the mobile phones. The centrally collected data can in turn be processed/classified and indexed in order to support efficient data-specific query resolution. The similar approach can be used to enable location tracking of mobile phones, which is discussed in Section 4.1

Among distributed approaches, location- and/or time-based query resolution has been studied extensively for traditional wireless sensor networks (WSNs), however the approaches presented are mainly for a network of static nodes. Thus, most of the location-specific indexing approaches are based on the fact that the data about an event happened at a particular location would be present at a sensor located at or nearby that location [12, 16]. Such an approach may not be useful for a smartphone based network, since the phone user may have moved to another location at the time of the query. We discuss the challenges associated with smartphone based networks in the next section.

3. RESEARCH CHALLENGES

The major challenge to locate the target mobile phones that were present in the given spatio-temporal window arises from the fact that the mobile phone (i.e. the data storage) may be moving in uncontrolled mobility pattern. This encourages the need to track the phone, which in turn leads to another problem: preserving identity and location privacy of the mobile phone users. We now discuss these challenges in detail.

Which are the target phones? If the location and the time duration of a historic event is known, then the first step involved in the query resolution process is to determine the set of phones that were present at that location at the given time. For an event that is currently happening, or that has just happened, it is easily possible to probe the access points, if they are available in the event region, which in turn can probe the nearby mobile phones. However, for older events, this approach may not be useful since the phone user may have moved to some other location.

Where are the phones located currently? If the nearby access points stored the information about all the phones that were in its range at different times, it would be possible to address the first challenge. However, it is crucial to understand the current locations of the phones, in order to access the required data from them, which will not be served unless the identity privacy of the phone is compromised.

How to reduce the search space? If some of the phones present at a given location have not captured the required data, they can be discarded from the search space. The third challenge focuses on finding the only subset of phones which may have actually stored the required information, in order to reduce the query resolution cost.

How to maintain location and identity privacy of the phone users? Allowing the system to track the mobile phone users, which were not a part of a particular sensing task or campaign, can raise major privacy concerns. Thus, it is crucial for the system to ensure location and identity privacy of the mobile phones. The privacy requirement negates the possibility of using a trivial solution to learn the current location of the target mobile phone, whereby the event-capturing mobile phone can just report its Id (e.g. its cellular contact number), along with its location and timestamp to a server present on the Internet via a nearby WiFi access point or the cellular data channel. The query-resolution would then just involve locating the mobile phone with the given Id, using the regular cellular infrastructure.

Note that these are the initial set of research challenges stemming from the underlying requirement. However, depending on the solution approach to be used to address this problem, additional set of, mainly systems-specific, requirements can arise, which are out of the scope of this paper.

4. ARCHITECTURAL ALTERNATIVES

Understanding the current location of mobile phones that were present at a given location at given time is crucial to support data retrieval from such nodes. This entails a need to have a suitable architectural support that can enable efficiently locating the target nodes in a scalable manner, without compromising the required privacy of mobile phone user(s). Based on these requirements, we present a design space in terms of possible architectural alternatives that can be used for smartphone-based sensor networks. We categorize the architectures into three types: centralized, semi-centralized, and distributed, depending on where the indexing structure is stored. While the centralized approach is predominantly used in the existing architectures for smartphone networks, the semi-centralized and distributed architectures are hardly explored. We overview the existing centralized architecture, and present a novel semi-centralized architecture in detail. We also suggest a solution approach for employing a distributed architecture for data retrieval.

4.1 Centralized Architecture

In a centralized approach, mobile phones can periodically update their current locations to a central server via nearby data collectors, such as WiFi access points, or by using a cellular data channel [11]. The server can in turn create a spatio-temporal index of the mobile phones' tracking data to figure out the phone that was present in the query-specific spatio-temporal window, and its current location. Infrastructural requirements for such networks are very similar to that of the existing centralized data collection networks [7, 8, 10, 11].

The centralized solutions are beneficial in that the whole indexing structure is available at one location with the information about which phone was located at which place at what time. Thus, it is possible to quickly figure out the set of target mobile phones that need to be contacted for successful query resolution. Furthermore, the server can act as an interface between the smartphone network and the end users, thereby providing a transparent access to the requested data without compromising the location and identity privacy of the target smartphones.

However, the benefits of the centralized approach come at a cost of significant overhead in terms of the bandwidth and energy usage on the mobile phones. First, each phone needs to invest its limited energy supply in performing localization periodically. Localization is a costly operation for a battery-operated mobile phone due to the involvement of radio-communication for it. Furthermore, while the GPS-based localization was observed to provide significantly better accuracy (up to 7 meters), the battery life observed (less than 7 hours) was considerably less than that of the WiFi or cell-towers based localization schemes, when the GPS was used continuously [10]. Second, the estimated location of the phone needs to be communicated via a WiFi access point or a cellular data channel, which again needs to use power-hungry radio communication. Additionally, the granularity of location updates of the mobile phones can significantly affect the overall battery life. Third, potentially all mobile phones that are being used as sensors may be used for this purpose, which will increase the back-haul network's bandwidth utilization considerably, irrespective of the utility of the updates.

4.2 Semi-centralized Architecture: MobiTrail

In order to address the bandwidth and energy-wastage problems with the centralized approaches, we propose a novel semi-centralized approach, *MobiTrail*, to efficiently locate the target mobile phones. The key idea is to store the index at an intermediate (access point) level. Essentially, when a mobile phone senses an event and stores the data locally, it notifies a nearby access point to initialize the trail. We term the access point as a Sensor Access Point (SAP), following the terminology used in the MetroSense Project [7]. The trail can be further maintained by adding entries at the nearby SAPs as the mobile phone moves away from the event location. Since, the SAPs could be associated with the public WiFi access points, it is critical to maintain the location and identity privacy of the mobile phones. Thus, the phone's actual location and its Id (e.g. cellular number) should never be stored at any SAP. If the trail-maintenance cost is prohibitive, techniques such as trail-compaction can be used. Furthermore, techniques to reduce the search space should be introduced to pre-eliminate the phones that do not store the query-specific data. We now discuss the MobiTrail approach in detail.

4.2.1 *Trail initialization*

When a phone senses new data, which can be learned by checking if the corresponding sensor is active or not, its intent behind the sensing activity is verified by checking if it was a part of any assigned task, as in the case of opportunistic sensing. If the sensing activity was not particularly tasked, the newly sensed information can be stored locally. Subsequently, a *trail-initialization* message containing the start and end timestamps, and location of the captured event is sent to a nearby SAP. If an SAP is not available in the nearby region, the notification is delayed until an SAP is encountered during the course of the mobile phone's mobility.

Note that the MobiTrail also needs to localize the phone, to in turn localize the event that it is capturing. However, the localization is performed only when a new event is sensed, and not periodically as in case of the centralized approach.

4.2.2 *Trail maintenance*

The phone's mobility trail is maintained in a similar way to a doubly-linked-list. Specifically, when a new SAP is encountered, the SAP notifies the previous (or preceding) SAP, and adds it as a next hop entry to its routing table to reach the trail-initializing SAP. The previous SAP adds the new SAP as a next hop entry for reaching the target mobile phone. Thus, the phone tracking operation can be performed by first sending the query to an SAP located in the desired event location, and following the trail afterwords to reach the target mobile phone(s). Note that, the query routing process on the SAP-level can take place in a regular manner as a packet routing process on an IP network.

If the phone-user is an active participant in the event capture, further optimizations to reduce the communication for trail-maintenance can be performed. For instance, if the event's importance is progressively decreasing over time, the granularity of trail-maintenance messages can be reduced accordingly.

4.2.3 Trail compaction

The complete trail can be removed if all of the sensed data is delivered to a central server/database as a part of a query response or manual uploading. Trail removal can be performed by traversing the trail backwards from the current location of the phone.

Furthermore, since SAPs are assumed to have a backend connection with the Internet, intermediate SAPs, whose job is to just redirect the query to the next SAP, can be eliminated from the trail. Thus, the compacted trail for each event can only have the first and the last SAP on the trail for that event. Similarly, if a longer trail contains multiple overlapping trails, each for a separate event, then the starting points of all the trails and the ending point must be maintained in the compacted version of the trail.

4.2.4 Reducing the search-space

If a query demands for a specific type of event, the search-space of target mobile phones can be reduced by collecting meta-information about the event at the closest SAP during the trail-initialization phase. The meta-information may include:

1. *Sensors parameters:* Sensors parameters can be used to check if the required sensory input has been captured. For instance, if a query is interested in a video footage of the event, the only mobile phones that had used their cameras can be tracked. Furthermore, if a query is looking for a specific quality of the event capture, other camera parameters such as the capturing-resolution, camera zoom, etc. can be used to deduce the event coverage quality.

2. *Event summaries:* The search space can be further reduced by supplying event summaries (e.g. wavelet based summaries [9]) to the nearest SAP during the trail-initialization phase. These summaries can be used to perform light-weight pre-matching for the query, to select and track the best matching mobile phones. For instance, all the mobile phones, who have captured the event data, may not be required in many cases, and thus the search-space reduction mechanism can be used to select the k most matching mobile phones to resolve the query.

4.2.5 Location and Identity Privacy

The fact that a phone is being tracked without having appropriate permissions from the phone user, may raise privacy concerns for the phone users. Thus, it is crucial for the system to ensure location and identity privacy of the mobile phones that it will track. Since, the query will be specific to the event-location, the actual location of the mobile phone capturing the event will never be exposed to the end users. For query resolution, the query needs to be sent to a SAP located near the event region, and never to the mobile phone who has captured the event data. Once the query reaches the SAP present in the query-specific region, it can follow the trail to reach the last SAP that can access the required data from the target mobile phone present in its communication range. If the target phone is not in the communication range of the last SAP on the trail, it needs to wait until it receives a message from the next SAP about the phone's presence in its communication range.

In order to maintain identity privacy, we propose to use event-specific identifiers, instead of phone-specific identifiers. Essentially, in the trail initialization phase, the SAP generates a unique event-id for the target phone, and uses it in its routing table, and also notifies it to the target phone. The event-id needs to be unique only for the SAP that generates it. The further SAPs on the trail utilize the same event-id, if it has not been used by them already. Otherwise, the SAP generates a new unique Id (unique specific to itself), and stores a mapping from the old Id to the new Id. If the mobile phone captures more than one events while moving, the corresponding SAPs store a mapping from multiple event-ids to a single unique event-id, and assign the new event-id to the phone.

It is critical to note that the unique Id generation and maintenance is a difficult, if not impossible, task in case of the centralized solution, because the Ids are event-specific and not phone-specific. Since, each phone can participate in sensing multiple events, potentially huge number of unique Ids need to be generated and maintained consistently at a global scale for a centralized scheme. MobiTrail avoids this problem by making use of the geographically distributed nature of the smart phones and SAPs. Thus, it is significantly simpler to maintain uniqueness at an individual SAP level, since the number of phones and the number of events that can potentially be in a range of a SAP could be very limited.

Although, data privacy is not a focus of this paper, user level control for tagging data as public or private coupled with traditional access control mechanisms can be explored [13].

4.2.6 Summary

In summary, MobiTrail provides a semi-centralized way to locate phones in a given spatio-temporal window. The advantages of MobiTrail are that it is a scalable, and privacy-preserving approach. In addition, it also supports optimizing the query resolution task by using the meta-information about the desired event, in terms of sensor parameters or event-data summaries, to pre-classify the most useful target mobile phones. Finally, the identity privacy ensuring mechanism is considerably easier than that in the centralized approach, which is based on exploiting the geographical separation of SAPs. The disadvantage of MobiTrail is that it has a slight overhead of maintaining the indexing structure at the SAP level, and the phone tracking process may take a little longer than the centralized approach.

4.3 Distributed Architecture

In a distributed architecture, indexing structure is stored directly on the mobile phones. The target mobile phone search problem is similar in nature with the search problem in peer-to-peer (P2P) networks. Thus, a distributed P2P lookup approach, such as Chord [15], can be adapted to solve the target mobile phone lookup problem, based on a given key representing the query-specified spatio-temporal window.

Note that the idea for distributed approach discussed here is in a preliminary stage, and needs more work to show its accuracy, while maintaining scalability and robustness for mobile phones case, which is a part of our future work. Advantages of using a distributed approach include scalability, and no need for the infrastructure to be in place to support query resolution. The disadvantages are more implementation-specific, and may include the lack of identity privacy for the phone users, and a possibly longer query resolution time in comparison with the centralized approach.

5. CONCLUSION AND FUTURE WORK

Smartphone based sensing is being used widely either as a part of the assigned sensing task, or simply out of the user's own interest. Majority of the existing work focused on sensing data as a part of the assigned task, and reporting it in real-time or opportunistically to the intended receiver(s). In this work, we focused on the cases where the sensed data is stored locally on the mobile phones. We identified a problem of efficiently locating the subset of mobile phones that have potentially captured data about the historical event of interest. We discussed research challenges associated with this problem, and presented a design space aimed at addressing the challenges. The design space provided a categorization of the architectural alternatives, including a novel semi-centralized approach, *MobiTrail*, to address this problem, along with their qualitative evaluations.

In future, we plan to quantitatively evaluate MobiTrail in simulations as well as on a testbed comprising of smartphones and SAPs. In addition, we plan to work on developing the distributed indexing structure for smartphones, whose solution approach is briefly discussed in the paper.

6. REFERENCES

[1] I. Akyildiz, T. Melodia, and K. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921–960, 2007.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[3] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory sensing. In *World Sensor Web Workshop*, pages 1–5. Citeseer, 2006.

[4] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, and R. Peterson. People-centric urban sensing. In *Proceedings of the 2nd annual international workshop on Wireless internet*, page 18. ACM, 2006.

[5] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, et al. The rise of people-centric sensing. *IEEE Internet Computing*, pages 12–21, 2008.

[6] S. Eisenman. *People-centric mobile sensing networks.* PhD thesis, Citeseer, 2008.

[7] S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, G. Ahn, and A. Campbell. Metrosense project: People-centric sensing at scale. In *First Workshop on World-Sensor-Web (WSW.2006).* Citeseer, 2006.

[8] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Ahn, and A. Campbell. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)*, 6(1), 2009.

[9] D. Ganesan, D. Perelyubskiy, and J. Heidemann. An evaluation of multi-resolution storage for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 89–102. ACM Press New York, NY, USA, 2003.

[10] S. Gaonkar, J. Li, R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: Sharing and querying content through mobile phones and social participation. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 174–186. ACM, 2008.

[11] A. Gupta, A. Kalra, D. Boston, and C. Borcea. MobiSoC: a middleware for mobile social computing applications. *Mobile Networks and Applications*, 14(1):35–52, 2009.

[12] X. Li, Y. Kim, R. Govindan, and W. Hong. Multi-dimensional range queries in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 63–75. ACM New York, NY, USA, 2003.

[13] Q. Liu, R. Safavi-Naini, and N. Sheppard. Digital rights management for content distribution. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21.* Australian Computer Society, Inc., 2003.

[14] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. ContextPhone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, pages 51–59, 2005.

[15] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, page 160. ACM, 2001.

[16] J. Winter and W. Lee. KPT: a dynamic KNN query processing algorithm for location-aware sensor networks. In *Proceeedings of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB 2004*, pages 119–124. ACM, 2004.