A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services*

Chi-Yin Chow
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN
cchow@cs.umn.edu

Mohamed F. Mokbel
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN
mokbel@cs.umn.edu

Xuan Liu
IBM Thomas J. Watson Research Center
Hawthorne, NY
xuanliu@us.ibm.com

ABSTRACT

This paper tackles a major privacy threat in current location-based services where users have to report their exact locations to the database server in order to obtain their desired services. For example, a mobile user asking about her nearest restaurant has to report her exact location. With untrusted service providers, reporting private location information may lead to several privacy threats. In this paper, we present a peer-to-peer (P2P) spatial cloaking algorithm in which mobile and stationary users can entertain location-based services without revealing their exact location information. The main idea is that before requesting any location-based service, the mobile user will form a group from her peers via single-hop communication and/or multi-hop routing. Then, the spatial cloaked area is computed as the region that covers the entire group of peers. Two modes of operations are supported within the proposed P2P spatial cloaking algorithm, namely, the *on-demand* mode and the *proactive* mode. Experimental results show that the P2P spatial cloaking algorithm operated in the on-demand mode has lower communication cost and better quality of services than the proactive mode, but the on-demand incurs longer response time.

Categories and Subject Descriptors: H.2.8 [Database Applications]: Spatial databases and GIS
General Terms: Algorithms and Experimentation.
Keywords: Mobile computing, location-based services, location privacy and spatial cloaking.

1. INTRODUCTION

The emergence of state-of-the-art location-detection devices, e.g., cellular phones, global positioning system (GPS) devices, and radio-frequency identification (RFID) chips results in a location-dependent information access paradigm, known as location-based services (LBS) [30]. In LBS, mobile users have the ability to issue location-based queries to the location-based database server. Examples of such queries include "where is my nearest gas station", "what are the restaurants within one mile of my location", and "what is the traffic condition within ten minutes of my route". To get the precise answer of these queries, the user has to provide her exact location information to the database server. With untrustworthy servers, adversaries may access sensitive information about specific individuals based on their location information and issued queries. For example, an adversary may check a user's habit and interest by knowing the places she visits and the time of each visit, or someone can track the locations of his ex-friends. In fact, in many cases, GPS devices have been used in stalking personal locations [12, 39]. To tackle this major privacy concern, three centralized privacy-preserving frameworks are proposed for LBS [13, 14, 31], in which a trusted third party is used as a middleware to blur user locations into spatial regions to achieve k-anonymity, i.e., a user is indistinguishable among other $k-1$ users. The centralized privacy-preserving framework possesses the following shortcomings: 1) The centralized trusted third party could be the system bottleneck or single point of failure. 2) Since the centralized third party has the complete knowledge of the location information and queries of all users, it may pose a serious privacy threat when the third party is attacked by adversaries.

In this paper, we propose a peer-to-peer (P2P) spatial cloaking algorithm. Mobile users adopting the P2P spatial cloaking algorithm can protect their privacy without seeking help from any centralized third party. Other than the shortcomings of the centralized approach, our work is also motivated by the following facts: 1) The computation power and storage capacity of most mobile devices have been improving at a fast pace. 2) P2P communication technologies, such as IEEE 802.11 and Bluetooth, have been widely deployed. 3) Many new applications based on P2P information sharing have rapidly taken shape, e.g., cooperative information access [9, 32] and P2P spatio-temporal query processing [20, 24].

Figure 1 gives an illustrative example of P2P spatial cloaking. The mobile user A wants to find her nearest gas station while being five anonymous, i.e., the user is indistinguishable among five users. Thus, the mobile user A has to look around and find other four peers to collaborate as a group. In this example, the four peers are B, C, D, and E. Then, the mobile user A cloaks her exact location into a spatial

*This work is supported in part by the Grants-in-Aid of Research, Artistry, and Scholarship, University of Minnesota.

ACM-GIS'06, November 10-11, 2006, Arlington, Virginia, USA.

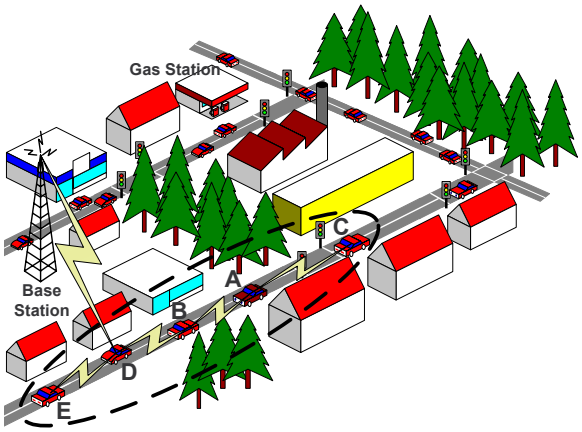


Figure 1: An example of P2P spatial cloaking

region that covers the entire group of mobile users A, B, C, D, and E. The mobile user A randomly selects one of the mobile users within the group as an *agent*. In the example given in Figure 1, the mobile user D is selected as an agent. Then, the mobile user A sends her query (i.e., what is the nearest gas station) along with her cloaked spatial region to the agent. The agent forwards the query to the location-based database server through a base station. Since the location-based database server processes the query based on the cloaked spatial region, it can only give a list of candidate answers that includes the actual answers and some false positives. After the agent receives the candidate answers, it forwards the candidate answers to the mobile user A. Finally, the mobile user A gets the actual answer by filtering out all the false positives.

The proposed P2P spatial cloaking algorithm can operate in two modes: *on-demand* and *proactive*. In the *on-demand* mode, mobile clients execute the cloaking algorithm when they need to access information from the location-based database server. On the other side, in the *proactive* mode, mobile clients periodically look around to find the desired number of peers. Thus, they can cloak their exact locations into spatial regions whenever they want to retrieve information from the location-based database server. In general, the contributions of this paper can be summarized as follows:

1. We introduce a distributed system architecture for providing anonymous location-based services (LBS) for mobile users.
2. We propose the first P2P spatial cloaking algorithm for mobile users to entertain high quality location-based services without compromising their privacy.
3. We provide experimental evidence that our proposed algorithm is efficient in terms of the response time, is scalable to large numbers of mobile clients, and is effective as it provides high-quality services for mobile clients without the need of exact location information.

The rest of this paper is organized as follows. Section 2 highlights the related work. The system model of the P2P spatial cloaking algorithm is presented in Section 3. The P2P spatial cloaking algorithm is described in Section 4. Section 5 discusses the integration of the P2P spatial cloaking algorithm with privacy-aware location-based database servers. Section 6 depicts the experimental evaluation of the P2P spatial cloaking algorithm. Finally, Section 7 concludes this paper.

2. RELATED WORK

The k-anonymity model [37, 38] has been widely used in maintaining privacy in databases [5, 26, 27, 28]. The main idea is to have each tuple in the table as k-anonymous, i.e., indistinguishable among other $k-1$ tuples. Although we aim for the similar k-anonymity model for the P2P spatial cloaking algorithm, none of these techniques can be applied to protect user privacy for LBS, mainly for the following four reasons: 1) These techniques preserve the privacy of the stored data. In our model, we aim not to store the data at all. Instead, we store *perturbed* versions of the data. Thus, data privacy is managed before storing the data. 2) These approaches protect the data not the queries. In anonymous LBS, we aim to protect the user who issues the query to the location-based database server. For example, a mobile user who wants to ask about her nearest gas station needs to protect her location while the location information of the gas station is not protected. 3) These approaches guarantee the k-anonymity for a snapshot of the database. In LBS, the user location is continuously changing. Such dynamic behavior calls for continuous maintenance of the k-anonymity model. (4) These approaches assume a unified k-anonymity requirement for all the stored records. In our P2P spatial cloaking algorithm, k-anonymity is a user-specified privacy requirement which may have a different value for each user.

Motivated by the privacy threats of location-detection devices [1, 4, 6, 40], several research efforts are dedicated to protect the locations of mobile users (e.g.,false dummies [23], landmark objects [18], and location perturbation [10, 13, 14]). The most closed approaches to ours are two centralized spatial cloaking algorithms, namely, the spatio-temporal cloaking [14] and the CliqueCloak algorithm [13], and one decentralized privacy-preserving algorithm [23]. The spatio-temporal cloaking algorithm [14] assumes that all users have the same k-anonymity requirements. Furthermore, it lacks the scalability because it deals with each single request of each user individually. The CliqueCloak algorithm [13] assumes a different k-anonymity requirement for each user. However, since it has large computation overhead, it is limited to a small k-anonymity requirement, i.e., k is from 5 to 10. A decentralized privacy-preserving algorithm is proposed for LBS [23]. The main idea is that the mobile client sends a set of false locations, called *dummies*, along with its true location to the location-based database server. However, the disadvantages of using dummies are threefold. First, the user has to generate realistic dummies to prevent the adversary from guessing its true location. Second, the location-based database server wastes a lot of resources to process the dummies. Finally, the adversary may estimate the user location by using cellular positioning techniques [34], e.g., the time-of-arrival (TOA), the time difference of arrival (TDOA) and the direction of arrival (DOA).

Although several existing distributed group formation algorithms can be used to find peers in a mobile environment, they are not designed for privacy preserving in LBS. Some algorithms are limited to only finding the neighboring peers, e.g., lowest-ID [11], largest-connectivity (degree) [33] and mobility-based clustering algorithms [2, 25]. When a mobile user with a strict privacy requirement, i.e., the value of $k-1$ is larger than the number of neighboring peers, it has to enlist other peers for help via multi-hop routing. Other algorithms do not have this limitation, but they are designed for grouping stable mobile clients together to facil-

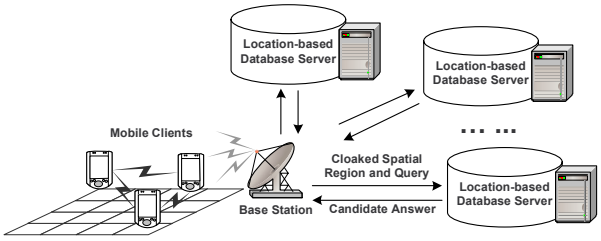


Figure 2: The system architecture

itate efficient data replica allocation, e.g., dynamic connectivity based group algorithm [16] and mobility-based clustering algorithm, called DRAM [19]. Our work is different from these approaches in that we propose a P2P spatial cloaking algorithm that is dedicated for mobile users to discover other $k-1$ peers via single-hop communication and/or via multi-hop routing, in order to preserve user privacy in LBS.

3. SYSTEM MODEL

Figure 2 depicts the system architecture for the proposed P2P spatial cloaking algorithm which contains two main components: *mobile clients* and *location-based database server*. Each mobile client has its own privacy profile that specifies its desired level of privacy. A privacy profile includes two parameters, k and A_{min}, k indicates that the user wants to be k-anonymous, i.e., indistinguishable among k users, while A_{min} specifies the minimum resolution of the cloaked spatial region. The larger the value of k and A_{min}, the more strict privacy requirements a user needs. Mobile users have the ability to change their *privacy profile* at any time. Our employed privacy profile matches the privacy requirements of mobiles users as depicted by several social science studies (e.g., see [4, 15, 17, 22, 29]).

In this architecture, each mobile user is equipped with two wireless network interface cards; one of them is dedicated to communicate with the location-based database server through the base station, while the other one is devoted to the communication with other peers. A similar multi-interface technique has been used to implement IP multi-homing for stream control transmission protocol (SCTP), in which a machine is installed with multiple network interface cards, and each assigned a different IP address [36]. Similarly, in mobile P2P cooperation environment, mobile users have a network connection to access information from the server, e.g., through a wireless modem or a base station, and the mobile users also have the ability to communicate with other peers via a wireless LAN, e.g., IEEE 802.11 or Bluetooth [9, 24, 32]. Furthermore, each mobile client is equipped with a positioning device, e.g., GPS or sensor-based local positioning systems, to determine its current location information.

4. P2P SPATIAL CLOAKING

In this section, we present the data structure and the P2P spatial cloaking algorithm. Then, we describe two operation modes of the algorithm: *on-demand* and *proactive*.

4.1 Data Structure

The entire system area is divided into grid. The mobile client communicates with each other to discover other $k-1$ peers, in order to achieve the k-anonymity requirement. The mobile client can thus blur its exact location into a cloaked spatial region that is the minimum grid area covering the $k-1$ peers and itself, and satisfies A_{min} as well. The grid area is represented by the ID of the left-bottom and right-top cells, i.e., (l, b) and (r, t). In addition, each mobile client maintains a parameter $\widehat{h}$ that is the required hop distance of the last peer searching. The initial value of $\widehat{h}$ is equal to one.

Algorithm 1 P2P Spatial Cloaking: Request Originator m

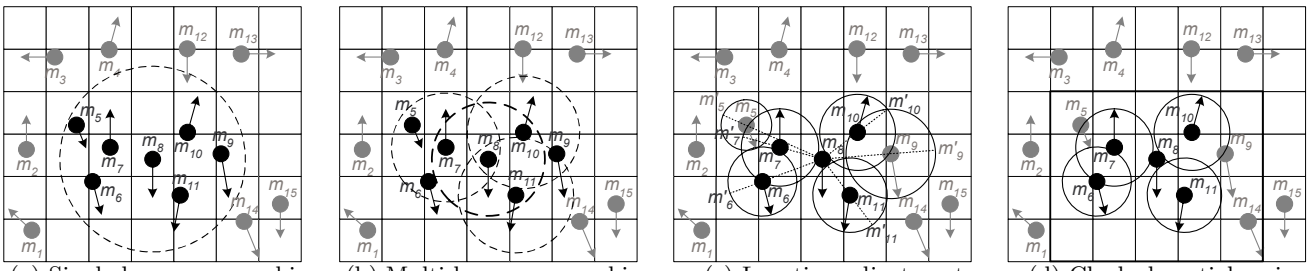
```
1: Function P2PCloaking-Originator (ĥ, k)
2: //Phase 1: Peer searching phase
3: The hop distance h is set to ĥ
4: The set of discovered peers T̂ is set to {∅}, and the number of
   discovered peers k̂ = |T̂| = 0
5: while k̂ < k − 1 do
6:    Broadcast a FORM_GROUP request with the parameter h (Al-
      gorithm 2 gives the response of each peer p that receives this
      request)
7:    T is the set of peers that respond back to m by executing
      Algorithm 2
8:    k̂ = |T|;
9:    if k̂ < k − 1 then
10:      if T = T̂ then
11:         Suspend the request
12:      end if
13:      h ← h + 1;
14:      T̂ ← T;
15:   end if
16: end while
17: //Phase 2: Location adjustment phase
18: for all T_i ∈ T do
19:    |mT_i.p'| ← the greatest possible distance between m and T_i.p
       by considering the timestamp of T_i.p's reply and maximum
       speed
20: end for
21: //Phase 3: Spatial cloaking phase
22: Form a group with k − 1 peers having the smallest |mp'|
23: ĥ ← the largest hop distance h_p of the selected k − 1 peers
24: Determine a grid area A that covers the entire group
25: if A < A_min then
26:    Extend the area of A till it covers A_min
27: end if
28: Randomly select a mobile client of the group as an agent
29: Forward the query and A to the agent
```

4.2 Algorithm

Figure 3 gives a running example for the P2P spatial cloaking algorithm. There are 15 mobile clients, m_1 to m_{15}, represented as solid circles. m_8 is the request originator, other black circles represent the mobile clients received the request from m_8. The dotted circles represent the communication range of the mobile client, and the arrow represents the movement direction. Algorithms 1 and 2 give the pseudo code for the *request originator* (denoted as m) and the *request receivers* (denoted as p), respectively. In general, the algorithm consists of the following three phases:

Phase 1: Peer searching phase. The request originator m wants to retrieve information from the location-based database server. m first sets h to $\widehat{h}$, a set of discovered peers T to $\{\emptyset\}$ and the number of discovered peers $\widehat{k}$ to zero, i.e., $|\mathcal{T}|$. (Lines 3 to 4 in Algorithm 1). Then, m broadcasts a FORM_GROUP request along with a message sequence ID and the hop distance h to its neighboring peers (Line 6 in Algorithm 1). m listens to the network and waits for the reply from its neighboring peers.

Algorithm 2 describes how a peer p responds to the FORM_GROUP request along with a hop distance h and a



(a) Single-hop peer searching (b) Multi-hop peer searching (c) Location adjustment (d) Cloaked spatial region

Figure 3: P2P spatial cloaking algorithm.

Algorithm 2 P2P Spatial Cloaking: Request Receiver p

```
1: Function P2PCloaking-Receiver (h)
2: //Let r be the request forwarder
3: if the request is duplicate then
4:     Reply r with an ACK message
5:     return;
6: end if
7: h_p ← 1;
8: if h = 1 then
9:     Send the tuple T =< p, (x_p, y_p), v_max_p, t_p, h_p > to r
10: else
11:    h ← h − 1;
12:    Broadcast a FORM_GROUP request with the parameter h
13:    T_p is the set of peers that respond back to p
14:    for all T_i ∈ T_p do
15:        T_i.h_p ← T_i.h_p + 1;
16:    end for
17:    T_p ← T_p ∪ {< p, (x_p, y_p), v_max_p, t_p, h_p >};
18:    Send T_p back to r
19: end if
```

message sequence ID from another peer (denoted as r) that is either the request originator or the forwarder of the request. First, p checks if it is a duplicate request based on the message sequence ID. If it is a duplicate request, it simply replies r with an ACK message without processing the request. Otherwise, p processes the request based on the value of h:

Case 1: $h = 1$. p turns in a tuple that contains its ID, current location, maximum movement speed, a timestamp and a hop distance (it is set to one), i.e., $< p, (x_p, y_p), v_{max_p}, t_p, h_p >$, to r (Line 9 in Algorithm 2).

Case 2: $h > 1$. p decrements h and broadcasts the FORM_GROUP request with the updated h and the original message sequence ID to its neighboring peers. p keeps listening to the network, until it collects the replies from all its neighboring peers. After that, p increments the h_p of each collected tuple, and then it appends its own tuple to the collected tuples $\mathcal{T}_p$. Finally, it sends $\mathcal{T}_p$ back to r (Lines 11 to 18 in Algorithm 2).

After m collects the tuples $\mathcal{T}$ from its neighboring peers, if m cannot find other $k-1$ peers with a hop distance of h, it increments h and re-broadcasts the FORM_GROUP request along with a new message sequence ID and h. m repeatedly increments h till it finds other $k-1$ peers (Lines 6 to 14 in Algorithm 1). However, if m finds the same set of peers in two consecutive broadcasts, i.e., with hop distances h and $h+1$, there are not enough connected peers for m. Thus, m has to relax its privacy profile, i.e., use a smaller value of k, or to be suspended for a period of time (Line 11 in Algorithm 1).

Figures 3(a) and 3(b) depict single-hop and multi-hop peer searching in our running example, respectively. In Figure 3(a), the request originator, m_8, (e.g., $k = 5$) can find $k-1$ peers via single-hop communication, so m_8 sets $h = 1$. Since $h = 1$, its neighboring peers, m_5, m_6, m_7, m_9, m_{10}, and m_{11}, will not further broadcast the FORM_GROUP request. On the other hand, in Figure 3(b), m_8 does not connect to $k-1$ peers directly, so it has to set $h > 1$. Thus, its neighboring peers, m_7, m_{10}, and m_{11}, will broadcast the FORM_GROUP request along with a decremented hop distance, i.e., $h = h - 1$, and the original message sequence ID to their neighboring peers.

Phase 2: ***Location adjustment phase***. Since the peer keeps moving, we have to capture the movement between the time when the peer sends its tuple and the current time. For each received tuple from a peer p, the request originator, m, determines the greatest possible distance between them by an equation, $|mp'| = |mp| + (t_c - t_p) \times v_{max_p}$, where $|mp|$ is the Euclidean distance between m and p at time t_p, i.e., $|mp| = \sqrt{(x_m - x_p)^2 + (y_m - y_p)^2}$, t_c is the current time, t_p is the timestamp of the tuple and v_{max_p} is the maximum speed of p (Lines 18 to 20 in Algorithm 1). In this paper, a conservative approach is used to determine the distance, because we assume that the peer will move with the maximum speed in any direction. If p gives its movement direction, m has the ability to determine a more precise distance between them.

Figure 3(c) illustrates that, for each discovered peer, the circle represents the largest region where the peer can locate at time t_c. The greatest possible distance between the request originator m_8 and its discovered peer, m_5, m_6, m_7, m_9, m_{10}, or m_{11} is represented by a dotted line. For example, the distance of the line $m_8m'_{11}$ is the greatest possible distance between m_8 and m_{11} at time t_c, i.e., $|m_8m'_{11}|$.

Phase 3: ***Spatial cloaking phase***. In this phase, the request originator, m, forms a virtual group with the $k-1$ nearest peers, based on the greatest possible distance between them (Line 22 in Algorithm 1). To adapt to the dynamic network topology and k-anonymity requirement, m sets $\widehat{h}$ to the largest value of h_p of the selected $k-1$ peers (Line 15 in Algorithm 1). Then, m determines the minimum grid area A covering the entire group (Line 24 in Algorithm 1). If the area of A is less than A_{min}, m extends A, until it satisfies A_{min} (Lines 25 to 27 in Algorithm 1). Figure 3(c) gives the $k-1$ nearest peers, m_6, m_7, m_{10}, and m_{11} to the request originator, m_8. For example, the privacy profile of m_8 is ($k = 5$, $A_{min} = 20$ cells), and the required cloaked spatial region of m_8 is represented by a bold rectangle, as depicted in Figure 3(d).

To issue the query to the location-based database server anonymously, m randomly selects a mobile client in the group as an agent (Line 28 in Algorithm 1). Then, m sends

the query along with the cloaked spatial region, i.e., A, to the agent (Line 29 in Algorithm 1). The agent forwards the query to the location-based database server. After the server processes the query with respect to the cloaked spatial region, it sends a list of candidate answers back to the agent. The agent forwards the candidate answer to m, and then m filters out the false positives from the candidate answers.

4.3 Modes of Operations

The P2P spatial cloaking algorithm can operate in two modes, *on-demand* and *proactive*.

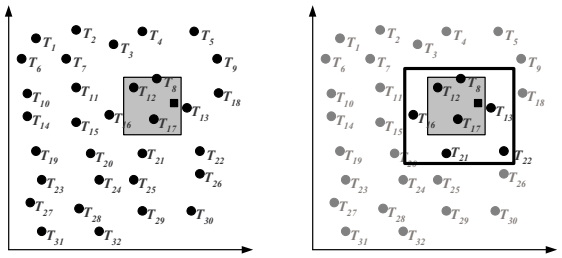
The *on-demand* mode: The mobile client only executes the algorithm when it needs to retrieve information from the location-based database server. The algorithm operated in the on-demand mode generally incurs less communication overhead than the proactive mode, because the mobile client only executes the algorithm when necessary. However, it suffers from a longer response time than the algorithm operated in the proactive mode.

The *proactive* mode: The mobile client adopting the proactive mode periodically executes the algorithm in background. The mobile client can cloak its location into a spatial region immediately, once it wants to communicate with the location-based database server. The proactive mode provides a better response time than the on-demand mode, but it generally incurs higher communication overhead and gives lower quality of service than the on-demand mode.

5. ANONYMOUS LOCATION-BASED SERVICES

Having the spatial cloaked region as an output form Algorithm 1, the mobile user m sends her request to the location-based server through an agent p that is randomly selected. Existing location-based database servers can support only exact point locations rather than cloaked regions. In order to be able to work with a spatial region, location-based servers need to be equipped with a privacy-aware query processor (e.g., see [29, 31]). The main idea of the privacy-aware query processor is to return a list of candidate answer rather than the exact query answer. Then, the mobile user m will filter the candidate list to eliminate its false positives and find its exact answer. The tighter the spatial cloaked region, the lower is the size of the candidate answer, and hence the better is the performance of the privacy-aware query processor. However, tight cloaked regions may represent relaxed privacy constrained. Thus, a trade-off between the user privacy and the quality of service can be achieved [31].

Figure 4(a) depicts such scenario by showing the data stored at the server side. There are 32 target objects, i.e., gas stations, T_1 to T_{32} represented as black circles, the shaded area represents the spatial cloaked area of the mobile client who issued the query. For clarification, the actual mobile client location is plotted in Figure 4(a) as a black square inside the cloaked area. However, such information is neither stored at the server side nor revealed to the server. The privacy-aware query processor determines a range that includes all target objects that are possibly contributing to the answer given that the actual location of the mobile client could be anywhere within the shaded area. The range is represented as a bold rectangle, as depicted in Figure 4(b). The server sends a list of candidate answers, i.e., T_8, T_{12}, T_{13}, T_{16}, T_{17}, T_{21}, and T_{22}, back to the agent. The agent next forwards the candidate answers to the requesting mobile client either through single-hop communication or through multi-hop routing. Finally, the mobile client can get the actual answer, i.e., T_{13}, by filtering out the false positives from the candidate answers.



(a) Server Side (b) Client Side

Figure 4: Anonymous location-based services

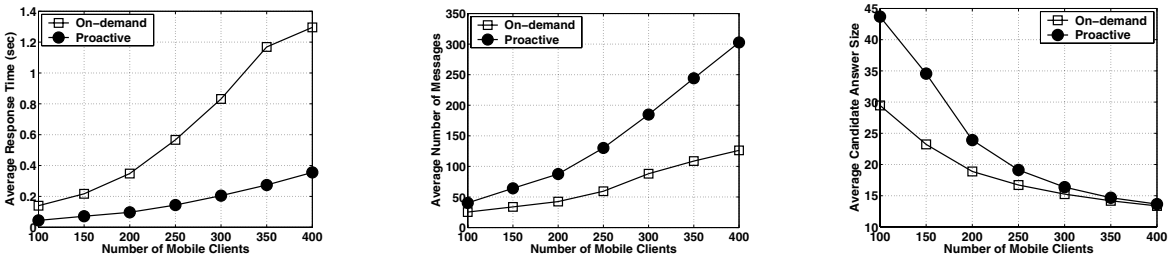
The algorithmic details of the privacy-aware query processor is beyond the scope of this paper. Interested readers are referred to [31] for more details.

6. EXPERIMENTAL RESULTS

In this section, we evaluate and compare the scalability and efficiency of the P2P spatial cloaking algorithm in both the *on-demand* and *proactive* modes with respect to the average response time per query, the average number of messages per query, and the size of the returned candidate answers from the location-based database server. The query response time in the *on-demand* mode is defined as the time elapsed between a mobile client starting to search $k-1$ peers and receiving the candidate answers from the agent. On the other hand, the query response time in the *proactive* mode is defined as the time elapsed between a mobile client starting to forward its query along with the cloaked spatial region to the agent and receiving the candidate answers from the agent. The simulation model is implemented in C++ using CSIM [35].

In all the experiments in this section, we consider an individual random walk model that is based on "random waypoint" model [7, 8]. At the beginning, the mobile clients are randomly distributed in a spatial space of $1,000 \times 1,000$ square meters, in which a uniform grid structure of 100×100 cells is constructed. Each mobile client randomly chooses its own destination in the space with a randomly determined speed s from a uniform distribution $U(v_{min}, v_{max})$. When the mobile client reaches the destination, it comes to a standstill for one second to determine its next destination. After that, the mobile client moves towards its new destination with another speed. All the mobile clients repeat this movement behavior during the simulation. The time interval between two consecutive queries generated by a mobile client follows an exponential distribution with a mean of ten seconds.

All the experiments consider one half-duplex wireless channel for a mobile client to communicate with its peers with a total bandwidth of 2 Mbps and a transmission range of 250 meters. When a mobile client wants to communicate with other peers or the location-based database server, it has to wait if the requested channel is busy. In the simulated mobile environment, there is a centralized location-based database server, and one wireless communication channel between the location-based database server and the mobile



(a) Response Time (b) Number of Messages (c) Candidate Answer Size

Figure 5: Number of mobile clients

clients, with a total bandwidth of 10 Mbps. We further assume that the size of the P2P communication message and the data record of a candidate answer is 64 bytes. Unless mentioned otherwise, we use a user privacy profile of $k = 5$ to 20 and $A_{min} = 0.01\%$ to 0.05% of the system area. The P2P spatial cloaking algorithm operated in the *proactive* mode is executed by the mobile client every 30 seconds.

6.1 Scalability

Figure 5 gives the scalability of the P2P spatial cloaking algorithm with respect to increasing the number of mobile clients from 100 to 400. In the experiment, although we only record the performance of the mobile clients in a $1,000 \times 1,000$ square meters area, there can be many other mobile clients located outside that area. Considering 400 mobile clients with a transmission range of 250 meters in the area, it gives a very crowded situation. Since the mobile clients generate queries independently, the number of concurrent queries in the system increases, when there are more mobile clients. Figure 5(a) depicts that the response time increases with larger number of mobile clients. As there are more mobile clients in the system, the number of concurrent queries increases that leads to higher network traffic; therefore, the transmission time of message passing gets longer.

The result also indicates that the algorithm operated in the *proactive* mode can effectively improve the response time, since the mobile client adopting the *proactive* mode need not spend any time on searching the required peers. However, the cost of the *proactive* mode is that it incurs larger number of messages and larger candidate answer size, as depicted in Figures 5(b) and 5(c), respectively. When the client population becomes more dense, it leads to larger number of messages broadcast by the mobile clients. Thus, the number of messages increases as the number of mobile clients gets larger. The mobile clients adopting the *proactive* mode execute the P2P spatial cloaking algorithm more frequent than the *on-demand* mode. Therefore, the *proactive* mode incurs larger number of messages than the *on-demand* mode.

The candidate answer size reduces when there are more mobile clients in the system. When the system becomes more dense, the mobile clients can generate smaller cloaked spatia regions, so the location-based database server can determine more accurate candidate answers, i.e., smaller false positives. The *proactive* mode offers a less precise location information of the peers than the *on-demand* mode. Thus, the greatest possible distance of the peers in the *proactive* mode is larger than that in the *on-demand* mode. As a result, the *proactive* mode generates larger cloaked spatial regions than the *on-demand* mode, i.e., the average candidate answer size of the *proactive* mode is larger than that of the *on-demand* mode.

6.2 Effect of Privacy Profile

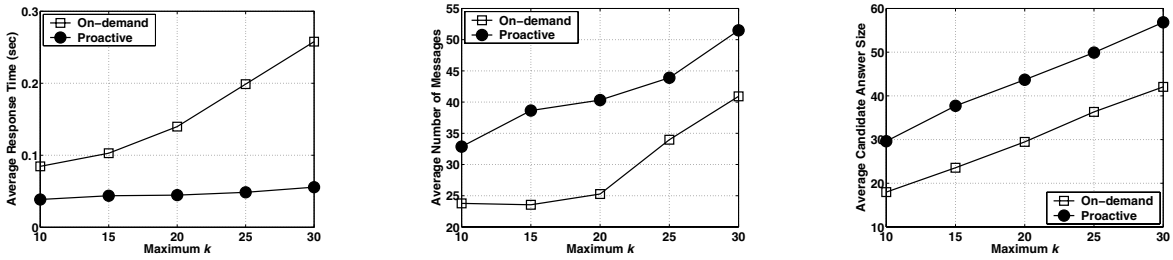
Figures 6 and 7 give the effect of more strict privacy profile on system performance by increasing the value of k and A_{min}, respectively. When the mobile clients increase the k anonymity parameter, they need to find more peers. In other words, they have to contact some peers that are farther away from them, in terms of hop distance, so the response time increases with a larger value of the k anonymity parameter, as depicted in Figure 6(a). The improvement of the *proactive* mode on the *on-demand* mode is the time taken on searching the required number of peers. The response time of the algorithm operated in the *proactive* mode slightly gets larger with increasing the value of k. Also, the hop distance between the agent and the querying mobile client is larger, so it takes longer time to forward the candidate answers from the agent to the requesting mobile client. With a larger value of k, the mobile clients have to broadcast more messages to their peers, so the number of messages increases as the k anonymity parameter gets larger, as depicted in Figure 6(b). Also, the candidate answer consists of more false positives, when the cloaked spatial region increases, so the candidate answer size gets larger with increasing the value of k (Figure 6(c)).

Figure 7 gives the results of increasing the A_{min} parameter, while the k anonymity parameter is from 5 to 10. Although the larger A_{min} does not lead to longer time taken on searching the required number of peers, it significantly increases the candidate answer size, as depicted in Figure 7(c), so the mobile clients experience longer latency in receiving the candidate answers from the agent. As a result, the response time slightly rises with increasing the value of A_{min} (Figure 7(a)). Since the P2P spatial cloaking algorithm operated in the *proactive* mode periodically searches the required number of peers, if the response time per query rises, the algorithm incurs larger number of messages per query, as depicted in Figure 7(b).

Similar to the experiment in scalability, the *proactive* mode incurs larger number of messages and larger candidate answer size than the *on-demand* mode.

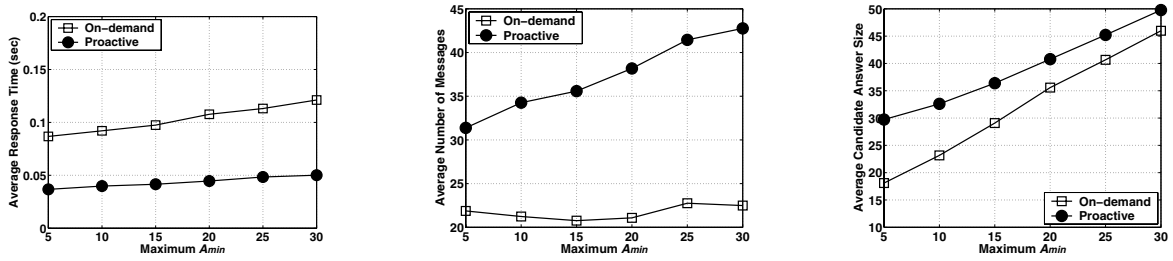
6.3 Effect of Client Disconnection

Client disconnection is one of the unique properties for mobile environments [3, 21], in which the mobile clients frequently disconnect themselves from the network voluntarily (to save energy) or involuntarily (handover or network failures). To evaluate the effect of client disconnection pattern on system performance, we consider 400 mobile clients with



(a) Response Time (b) Number of Messages (c) Candidate Answer Size

Figure 6: Maximum k (minimum $k = 5$)



(a) Response Time (b) Number of Messages (c) Candidate Answer Size

Figure 7: Maximum A_{min} (minimum $A_{min} = 0.01$)

various client disconnection probabilities, from 0.05 to 0.3. After a mobile client finishes a query, there is a probability that it will become disconnected from the network for five seconds.

Figure 8(a) gives that the response time slightly decreases with increasing the client disconnection probability. This is because the mobile clients can generally find enough number of peers via single-hop communication, even though the disconnection probability is equal to 0.3. Thus, when there are more mobile clients disconnected from the network, there is less network traffic that leads to a shorter response time. When a mobile client broadcasts a FORM_GROUP request, there are fewer peers receiving the message or returning tuples to it with increasing the client disconnection probability, so the P2P spatial cloaking algorithm incurs smaller number of messages, as depicted in Figure 8(b).

Figure 8(b) also gives that the number of messages for the algorithm operated in the *proactive* mode drops more rapidly than that for the *on-demand* mode with respect to increasing the client disconnection probability from 0.05 to 0.2. When the mobile clients adopting the *proactive* mode disconnect from the network, the P2P spatial cloaking algorithm is blocked. After the mobile clients re-connect to the network, they execute the algorithm to issue queries to the location-based database server. Then, the mobile clients will disconnect from the network again. Therefore, the number of messages of the *proactive* mode decreases with increasing the disconnection probability. Figure 8(c) depicts that the candidate answer size gets larger with increasing the disconnection probability. When there are more peers disconnected from the network, the mobile client may need to select some farther peers to satisfy its privacy requirement. As a result, the mobile client has a larger cloaked spatial region that incurs more false positives included in the candidate answers, i.e., the candidate answer size increases.

6.4 Accuracy

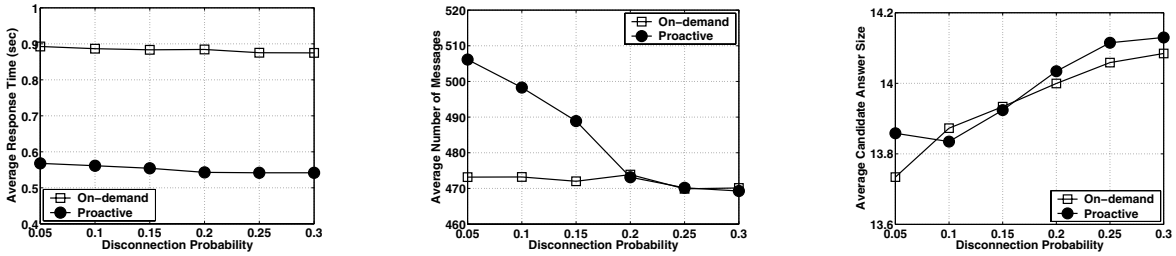
Since the querying mobile client keeps moving when the location-based database sever is processing its query, the candidate answers may not contain the correct answer, i.e., the nearest target object, to the querying mobile client. For all the previous experiments, we record that the accuracy of the candidate answers is over 99.9 percent on average. Thus, the P2P spatial cloaking algorithm provides very high quality anonymous location-based services for the mobile clients.

Table 1: The accuracy of the candidate answers

	On-demand	Proactive
Scalability	99.99%	100%
Effect of k	99.92%	99.95%
Effect of A_{min}	99.97%	99.98%
Effect of Client Disconnection	100%	100%

7. CONCLUSION

This paper introduces a P2P spatial cloaking algorithm that allows the mobile user to entertain anonymous location-based services without the help of any centralized third parties. By using the algorithm, the mobile user can find the required number of peers to form a group and then she determines the minimum grid area that satisfies her privacy requirements. After that, the mobile user randomly selects a peer from the group as an agent. The agent is responsible for communicating with the location-based database server and forwarding the answer to the querying mobile user. The P2P spatial cloaking algorithm can operate in two modes, *on-demand* and *proactive*. For the on-demand mode, the mobile user only executes the algorithm when she needs to access information from the location-based database server. The mobile user adopting the proactive mode periodically executes the algorithm in background, so the mobile user can cloak her exact location into a spatial region whenever she needs to enlist the location-based database server for help. Experimental evaluation studies the algorithm operated in both the on-demand and proactive modes and shows that there is a performance trade-off between them. The



(a) Response Time (b) Number of Messages (c) Candidate Answer Size

Figure 8: Client disconnection probability

algorithm operated in the proactive mode outperforms the on-demand mode in terms of the response time, but it generally incurs higher communication overhead and gives lower quality of service than the on-demand mode.

8. REFERENCES

[1] L. Ackerman, J. Kempf, and T. Miki. Wireless Location Privacy: A Report on Law and Policy in the United States, the European Union, and Japan. Technical Report DCL-TR2003-001, DoCoMo Commuinication Laboratories, USA, 2003.

[2] B. An and S. Papavassiliou. A Mobility-based Clustering Approach to Support Mobility Management and Multicast Routing in Mobile Ad-hoc Wireless Networks. *International Journal of Network Management*, 11(6):387–395, 2001.

[3] D. Barbará and T. Imielinski. Sleepers and Workaholics: Caching Strategies in Mobile Environments. *The International Journal on VLDB*, 4(4):567–602, 1995.

[4] L. Barkhuus and A. K. Dey. Location-Based Services for Mobile Telephony: A Study of Users' Privacy Concerns. In *Proceeding of the IFIP Conference on Human-Computer Interaction, INTERACT*, 2003.

[5] R. J. Bayardo Jr. and R. Agrawal. Data Privacy through Optimal k-Anonymization. In *ICDE*, 2005.

[6] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[7] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *MobiCom*, 1998.

[8] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications and Mobile Computing (WCMC)*, 2(5):483–502, 2002.

[9] C.-Y. Chow, H. V. Leong, and A. T. S. Chan. Distributed Group-based Cooperative Caching in a Mobile Broadcast Environment. In *MDM*, 2005.

[10] M. Duckham and L. Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. In *Pervasive*, 2005.

[11] A. Ephremides, J. Wieselthier, and D. J. Baker. A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling. *Proceedings of IEEE*, 75(1):56–73, 1987.

[12] Foxs News. Man Accused of Stalking Ex-Girlfriend With GPS. http://www.foxnews.com/story/0,2933,131487,00.html. Sep 04, 2004.

[13] B. Gedik and L. Liu. A Customizable k-Anonymity Model for Protecting Location Privacy. In *ICDCS*, 2005.

[14] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *MobiSys*, 2003.

[15] M. Gruteser and X. Liu. Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security and Privacy*, 2(2):28–34, 2004.

[16] T. Hara. Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility. In *INFOCOM*, 2001.

[17] U. Hengartner and P. Steenkiste. Protecting Access to People Location Information. In *Proceeding of the International Conference on Security in Pervasive Computing, SPC*, 2003.

[18] J. I. Hong and J. A. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *In Proceedings of The International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2004.

[19] J.-L. Huang, M.-S. Chen, and W.-C. Peng. Exploring Group Mobility for Replica Data Allocation in a Mobile Environment. In *CIKM*, 2003.

[20] Z. Huang, C. S. Jensen, H. Lu, and B. C. Ooi. Skyline Queries Against Mobile Lightweight Devices in MANETs. In *ICDE*, 2006.

[21] T. Imielinski and B. R. Badrinath. Mobile Wireless Computing: Challenges in Data Management. *Communications of the ACM (CACM)*, 37(10):18–28, 1994.

[22] E. Kaasinen. User Needs for Location-aware Mobile Services. *Personal and Ubiquitous Computing*, 7(1):70–79, 2003.

[23] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique using Dummies for Location-based Services. In *Proceedings of IEEE International Conference on Pervasive Services, ICPS*, 2005.

[24] W.-S. Ku, R. Zimmermann, H. Wang, and C.-N. Wan. Adaptive Nearest Neighbor Queries in Travel Time Networks. In *GIS*, 2005.

[25] G. H. K. Lam, H. V. Leong, and S. C. F. Chan. GBL: Group-Based Location Updating in Mobile Environment. In *DASFAA*, 2004.

[26] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian Multidimensional k-Anonymity. In *ICDE*, 2006.

[27] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient Full-Domain k-Anonymity. In *SIGMOD*, 2005.

[28] A. Meyerson and R. Williams. On the Complexity of Optimal k-Anonymity. In *PODS*, 2004.

[29] M. F. Mokbel. Towards Privacy-Aware Location-Based Database Servers. In *Proceedings of the ICDE International Workshop on Privacy Data Management, PDM*, 2006.

[30] M. F. Mokbel, W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Towards Scalable Location-aware Services: Requirements and Research Issues. In *GIS*, 2003.

[31] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query Procesing for Location Services without Compromising Privacy. In *VLDB*, 2006.

[32] M. Papadopouli and H. Schulzrinne. Effects of Power Conservation, Wireless Coverage and Cooperation on Data Dissemination among Mobile Devices. In *MobiHoc*, 2001.

[33] A. K. Parekh. Selecting Routers in Ad-Hoc Wireless Network. In *Proceedings of the International Telecommunications Symposium*, 1994.

[34] J. Reed, K. Krizman, B. Woerner, and T. Rappaport. An Overview of the Challenges and Progress in Meeting the E-911 Requirement for Location Service. *IEEE Personal Communications Magazine*, 5(3):30–37, 1998.

[35] H. Schwetman. *User's Guide CSIM19 Simulation Engine (C++ Version)*. Mesquite Software Inc.

[36] R. Stewart and Q. Xie. *Stream Control Transmission Protocol (SCTP): a reference guide*. Addison-Wesley Publishing Company, Boston, 2001.

[37] L. Sweeney. Achieving k-Anonymity Privacy Protection using Generalization and Suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[38] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[39] USAToday. Authorities: GPS System Used to Stalk Woman. http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm, Dec 30, 2002.

[40] J. Warrior, E. McHenry, and K. McGee. They Know Where You Are . *IEEE Spectrum*, 40(7):20–25, 2003.