Cloaking Locations for Anonymous Location Based Services: A Hybrid Approach

Chengyang Zhang and Yan Huang

Computer Science and Engineering Department
University of North Texas

Abstract. An important privacy issue in Location Based Services(*LBS*) is to hide a user's identity while still provide quality location based services. Previous work has addressed the problem of locational $\mathcal{K}$-anonymity either based on centralized or decentralized schemes. However, a centralized scheme relies on an anonymizing server(*AS*) for location cloaking, which may become the performance bottleneck when there are large number of clients. More importantly, holding information in a centralized place is more vulnerable to malicious attacks. A decentralized scheme depends on peer communication to cloak locations and is more scalable. However, it may pose too much computation and communication overhead to the clients. The service fulfillment rate may also be unsatisfied especially when there are not enough peers nearby.

This paper proposes a new hybrid framework called *HiSC* that balances the load between the *AS* and mobile clients. *HiSC* partitions the space into base cells and a mobile client claims a surrounding area consisting of base cells. The number of mobile clients in the surrounding cells is kept and updated at both client and *AS* sides. A mobile client can either request cloaking service from the centralized *AS* or use a peer-to-peer approach for spatial cloaking based on personalized privacy, response time, and service quality requirements. *HiSC* can elegantly distribute the work load between the *AS* and the mobile clients by tuning one system parameter *base cell size* and two client parameters - *surrounding cell size* and *tolerance count*.

By integrating salient features of two schemes, *HiSC* successfully preserves query anonymity and provides more scalable and consistent service. Both the *AS* and the clients can enjoy much less work load. Additionally, we propose a simple yet effective RRS(random range shifting)algorithm to prevent possible privacy leakage that would exist in the original P2P approach.

Our experiments show that *HiSC* can elegantly balance the work load based on privacy requirements and client distribution. *HiSC* provides close to optimal service quality. Meanwhile, it reduces the response time by more than an order of magnitude from both the P2P scheme and the centralized scheme when anonymity level(value of $\mathcal{K}$) or number of clients is large. It also reduces the update message cost of the *AS* by nearly 6 times and the peer searching message cost of the clients by more than an order of magnitude.

1 Introduction

Location-based services provide mobile users personalized services from their current locations using one of several positioning technologies, e.g. GPS, cell-phone positioning, and positioning through Wi-Fi access points. Examples of location based services include wireless 911 emergency service, traffic advisories, location-aware advertising, tourist services, location-based games, and navigation. Other services can combine personal preference information with present locations to help users find food, lodging, and entertainment fitting their tastes and pocketbooks. While enjoying the convenience brought by these new applications, people start to worry about a new type of privacy threat, namely, the privacy issues introduced by the releasing of location information to untrusted third parties. Location information is sensitive because it is ubiquitous and can lead to many other information. For example, the frequently visited place such as a hospital may release a person's health condition. Through commonly visited places, people can be linked together. The risks of locational privacy breach range from releasing information about visits to sensitive places to enabling unwanted virtual or physical stalking. Since positioning touches upon delicate privacy issues (checking where a person is), strict ethics and privacy measures are strongly recommended for services that use positioning.

Policy enforcement approaches [1–5] require that a user should be informed about location privacy policies and explicitly give consent to a service provider before the service starts. Technological approaches attempt to tackle this major privacy concern by anonymizing service requesters' location information sent to the location based server while still provide quality location based services. The privacy is generally measured by $\mathcal{K}$-anonymity[6, 7] requirements. Two main approaches have been proposed for location anonymizing: centralized [8–13] and decentralized [14–19].

A centralized approach utilizes a centralized trusted anonymizing server (AS) to anonymize a user's request. The AS strips off a service requester's identifying information, such as user's network address and name. Then the AS requests service from the location based service (LBS) provider. However, stripping of such information is not enough because location itself may release sensitive information. For example, some private location, e.g. a house, exclusively belongs to a single person or a small group of people. If a user makes service requests frequently enough, his/her trajectory can be easily traced. When the trajectory contains such private location, the whole trajectory can be associated with the user. Thus an important task for the AS in the centralized approach is to anonymize the location information of the service requester. In current literature, AS anonymizes location by either spatial/temporal cloaking or path confusion. In both schemes, the AS has to keep track of the positions of all the moving objects and performs anonymizing for each service request. As a result, if the server is hacked, all the private information on the server will be released. More importantly, the risk of information leakage is high when the information is held in a centralized place. Hence, one disadvantage of the centralized scheme is that the AS may become the target of attack. Moreover, to provide acceptable level

of service quality, the *AS* has to maintain client counts(i.e. number of mobile clients) information in a fine spatial resolution. This can easily lead the *AS* to become a performance bottleneck when there are large number of client updates and requests. Therefore the *AS* may have to improve performance with the price of lower service quality.

A decentralized approach can achieve anonymity without an *AS*. One of the methods in this category utilizes peer-to-peer communications. Each moving object probes its neighborhood to look for other moving objects and anonymizes its own location using the information collected. In general, this method has higher service quality. However, it does not guarantee that the service requests can be fulfilled because there may not be enough peers nearby. As a result, the service availability is not consistent. For hand held devices with limited capabilities, a peer-to-peer system may also pose too much computational and communicational overhead.

In most of the approaches discussed above, the anonymizing process cannot be tuned by the user. The user has to accept the preset level of response time, service fulfill rate, or service quality.

This paper proposes a new hybrid framework called *HiSC* that balances the load between the *AS* and mobile users. *HiSC* partitions the space into cells. The number of mobile clients in the surrounding cells is kept and updated at both client and the *AS* sides. A mobile client can either request cloaking service from the centralized *AS* or use peer-to-peer approach for spatial cloaking based on personalized privacy, response time, and service precision requirements. The major contributions of the paper are:

- *HiSC* effectively preserves query anonymity even if all the location information is disclosed.
- *HiSC* can elegantly distribute the work load among all parties in the system. Using lazy update mechanism(i.e., the clients only update their locations when they move across cell boundaries), the *AS* can enjoy much less work load caused by location updates.
- *HiSC* can provide more scalable and consistent service. It guarantees service availability even if the clients are sparsely distributed. Meanwhile, a client may adjust two simple parameters(*surrounding cell size* and *tolerance count*), which may be combined with privacy parameter($\mathcal{K}$) and system parameter *base cell size* to meet personalized response time and service quality requirements.
- Additionally, a simple yet effective RRS(random range shifting) algorithm is designed to prevent possible privacy leakage that would exist in the original P2P approach.
- Our experiments show that *HiSC* provides close to optimal service quality. Meanwhile, it reduces response time by more than an order of magnitude from both the P2P scheme and the centralized scheme when anonymity level(value of $\mathcal{K}$) is high or number of clients is large. It also reduces the update message cost of the *AS* by nearly 6 times and the peer searching message cost of the clients by more than an order of magnitude.

The rest of the paper is organized as follows. Section 2 describes related work. The system architecture and the mechanisms of centralized and decentralized approaches are outlined in Section 3. Section 4 presents our *HiSC* scheme. This section emphasizes our major improvements, including the random range shifting algorithm. Section 5 discusses the algebraic cost models. Experimental validation results are presented in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

Technological approaches for protecting locational privacy are often centered around the notion of $\mathcal{K}$-anonymity that was first introduced in [6]. A $\mathcal{K}$-anonymity is defined as "A release provides $\mathcal{K}$-anonymity protection if the information for each person contained in the release cannot be distinguished from at least $\mathcal{K}-1$ individuals whose information also appears in the release." This model was used in several database applications, such as [20–22]. The first attempt to extend $\mathcal{K}$-anonymity to locational $\mathcal{K}$-anonymity was proposed in [8]. In [8], the disclosed location of a requester is expanded to an area that includes at least $\mathcal{K}-1$ other mobile users. Then any of the $\mathcal{K}$ people within the disclosed area could have been the user. For private locations that are likely to release a requester's identification, the cloaking area is generally large using $\mathcal{K}$-anonymity due to small number of users in privacy areas. This helps in protecting location privacy.

Based on the methods proposed, the research efforts dedicated to this area may fall into two categories: centralized and decentralized schemes.

In a centralized approach, service requests are first transmitted to a third-party trusted server called location anonymizing server(AS). Various perturbation operations can be performed on the AS to obfuscate the original location information. Location based service(LBS) requests are then issued from the AS to the LBS providers. The AS also filters the query results and returns the exact answers to the original requesters.

In [9] a *CliqueCloak* algorithm is used for cloaking, aiming at avoiding or reducing known location privacy threats before forwarding requests to LBS providers. The infrastructure in [23] delays and reorders messages from subscribers within a mixed zone to confuse an observer. The path confusion algorithm in [12] tries to perturbate crossing paths in areas where at least two users meet. This increases the chances that an adversary would confuse the paths of different users.

The Casper [11] consists of two main components, the AS and the privacy-aware query processor. The AS blurs a user's exact location information into a cloaking spatial region based on user specified privacy requirements. The privacy-aware query processor is embedded inside the AS to deal with the cloaking spatial areas rather than exact location information. This framework features a quad-tree data structure that maps the location information into grids with different resolutions. The cloaking algorithm goes through the quad-tree in a bottom-up fashion to find a spatial region that meets the privacy requirements. This approach also requires the AS to dynamically keep track of the locations

of mobile devices in a fine spatial resolution. This can easily lead the *AS* to become a performance bottleneck when there are large number of client updates and requests. Due to the limitations of the quad-tree structure, the calculated cloaking region is often larger than required, which may cause lower service quality.

On the other hand, a decentralized approach does not involve any *AS*. For example, the paper [19] lets the mobile clients generate false locations and send them along with the real locations to LBS. For every location update, a user would send n different locations to the server. Only one of them is true. The rest are dummies. Thus, the server cannot know which location is the actual one. However, it is still possible to detect false dummies through data mining techniques if the algorithm used for dummy generation is not selected appropriately.

A peer-to-peer spatial cloaking algorithm is proposed in [18]. A mobile client first finds a peer group that meets the privacy requirement(i.e. find $\mathcal{K}-1$ neighbors) through peer searching. Then the client calculates a region that includes the $\mathcal{K}-1$ peers. It randomly chooses one of its neighbors as the agent to request *LBS* using the cloaking region. The query results are eventually forwarded to the original client through the agent. Unfortunately, this approach cannot guarantee service availability when clients are sparsely distributed. Moreover, privacy leakage may happen because the requester tends to be in the center of the cloaking region. In [14, 15], a decentralized approach based on Hilbert Curve is proposed to meet the reciprocity property requirements of the location cloaking algorithm. This approach guarantees the query anonymity even location information is disclosed to the adversary. However, each client needs to maintain relatively complex data structure and communication protocol as well as long range communication among peers. Therefore additional computation and communication cost may be posed to clients with limited capabilities.

In general, the *AS* in the centralized approach may become the performance bottleneck and target of attack. The service quality is often at a coarse level. This can be alleviated by using decentralized/distributed *AS*, the extreme case of which would be a P2P approach. On the other hand, the P2P approach has the limitation in providing consistent service regardless of client distributions, which can be remedied by introducing a trusted server. Our proposed *HiSC* approach integrates the salient features of the two approaches and presents a new solution to the problem.

3 System Architecture and Background

The system architecture of *HiSC* is shown in Figure 1. It consists of three major components: mobile clients, an *AS*, and *LBS* servers. Mobile clients communicate with each other via wireless LAN protocols, e.g. 802.11e or bluetooth. The clients can also communicate with the *AS* and the *LBS* servers through a base station. The *AS* is trusted, and has knowledge about a mobile client's identifier, service requests, and exact location. The *LBS* servers are not trusted. They do not have knowledge of the identifier of a mobile client who requests a particular service.

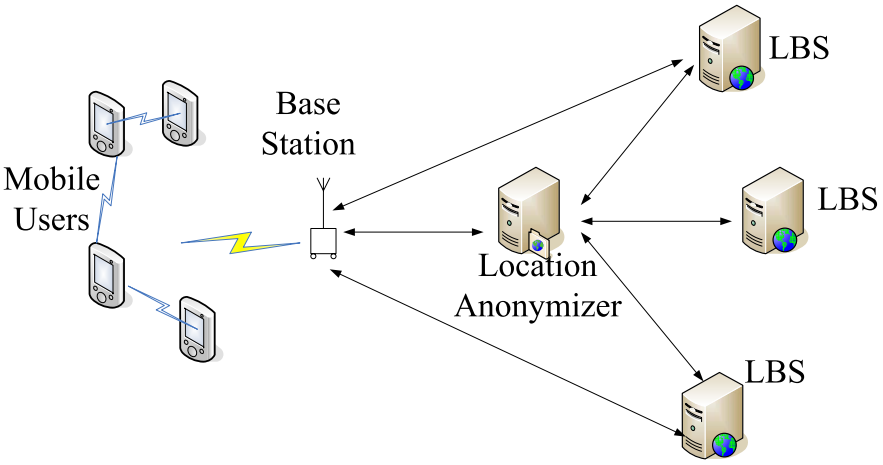


Fig. 1. System Overview

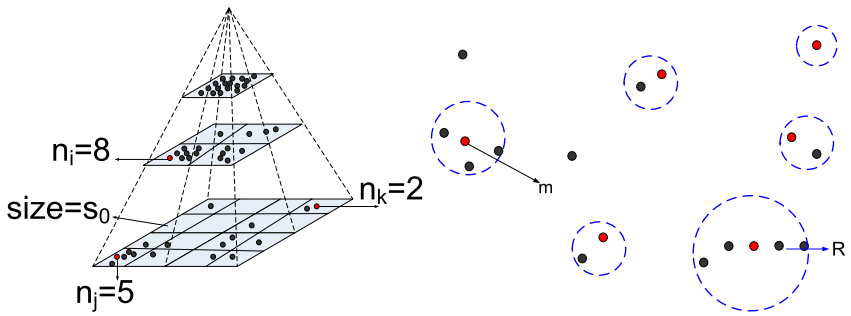
They do not know the precise location of the service requester and only know the request comes from a cloaking region - a region that contains at least $\mathcal{K}$ mobile clients. This will prevent location based identifier leakage.

The threat model behind the *HiSC* is that an adversary may combine the location information with a user's *LBS* requests to identify the requester's identity. Location information itself may not be sensitive. Therefore even all the location information is disclosed, the adversary still cannot deduce a person's identity if his/her query is cloaked in a $\mathcal{K}$-anonymity set.

Based on privacy requirements and current client counts information, spatial cloaking is handled by either the *AS* or mobile clients using two complementary schemes. The centralized scheme delegates the task to the *AS* and the P2P based scheme fulfills the task through peer to peer communication. We briefly discuss the cloaking process in the centralized scheme and the P2P based scheme before presenting our *HiSC* approach.

3.1 Centralized Scheme

In the centralized scheme[11], all the *LBS* service requests are first submitted to the *AS* for spatial cloaking before they are forwarded to a *LBS* provider. The *AS* strips off the identifier as well as cloaks the location of the requester. It also maintains a quad tree data structure to keep track of the counts(number) of mobile clients at different spatial resolutions, as shown in Figure 2 (a). Each cell in the quad tree stores number of mobile clients in that cell. When it receives a cloaking request from a client at location l, the *AS* uses the quad tree to find the smallest cell c that contains at least $\mathcal{K}$ mobile clients and use c to request *LBS* for the mobile client. The responses from the *LBS* provider are filtered according to location l and returned to the mobile client.



(a) Centralized Scheme (b) P2P Based Scheme

Fig. 2. Scheme illustration

The *AS* cannot use a cloaking region smaller than the size of the base cell (bottom level cell) s_0. Therefore s_0 determines the best cloaking quality that the *AS* can provide. s_0 need to be small enough to provide acceptable quality of service. Meanwhile, the maintenance of the quad tree data structure on the *AS* requires continuous and frequent location update from the mobile clients. Each client sends the *AS* its updated location information periodically, e.g. every 30 seconds. An efficient algorithm is necessary to maintain the quad tree structure. Note that even if there is no *LBS* request, the *AS* still has to carry on the heavy work load of keeping track of location updates. The *AS* may become a bottleneck when the base cell size is small, location update rate is high, or too many clients are requesting *LBS* at the same time.

In the proposed *HiSC* approach, however, we use a lazy update mechanism, i.e., the clients only update their locations when they move across cell boundaries. Moreover, the cloaking region may be smaller than s_0 because cloaking is performed using P2P communication if the clients are too dense. Therefore we can increase the size of s_0 to reduce update load without losing quality of service.

3.2 P2P based Scheme

In the P2P based scheme[18], spatial cloaking is performed using pure P2P communication. Through a multi-hop peer searching process, a client tries to find a group of at least $\mathcal{K}$ peers shown as circles in Figure 2 (b). A client will use the circle as the cloaking region to request *LBS*.

Note that if clients are sparsely distributed, a mobile client may not be able to find $\mathcal{K}-1$ other mobile clients and thus the requirement of $\mathcal{K}$-anonymity may not be fulfilled. This is due to the limitation of a mobile client's communication range. For example in Figure 2 (b), for $\mathcal{K}=5$, only service requests from clients in circle R can be fulfilled. One solution of this problem is to wait for some time and try again. This may result in substantial latency. To reduce the response latency

caused by the peer searching process, the clients may periodically calculate the cloaking region proactively. This however, may incur substantial communication overhead. Another potential problem of the P2P based scheme is that the client tends to be in the center of the cloaking region. Even if the identifier of a requester is unknown, the association of a request with a location may be inferred if the precise location information is disclosed [14], which may lead to privacy leakage. By contrast, in our *HiSC* approach, service can almost always be fulfilled because the *AS* can help to find the cloaking region when the clients are too sparse. We also propose a simple yet effective random range shifting algorithm to deal with the privacy leakage problem.

4 The Proposed Hybrid Spatial Cloaking Approach (*HiSC*)

The goal of the proposed *HiSC* approach is to flexibly distribute the work of spatial cloaking between mobile clients and the *AS*. Utilizing complementary features of the centralized and P2P-based schemes, *HiSC* can achieve optimized system performance in terms of response time, service quality, server capacity, and client computation and resource capacity.

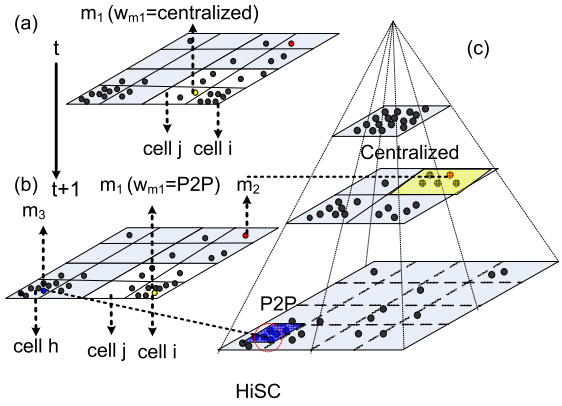


Fig. 3. Hybrid Approach Illustration ($K_m = 5, s_m = 1$ for any mobile client m)

HiSC can elegantly switch between different working modes by tuning a very small number of parameters. Specifically, each mobile client m can choose its more preferred working mode - Centralized, Direct, or P2P, by adjusting two parameters: the *surrounding cell size* s_m[1] and the *tolerance count* ϵ_m. Here the *tolerance count* ϵ_m refers to a threshold specifying extra number of clients from

[1] For writing convenience, we will use s_m to represent both m's surrounding cell and the size of the cell.

K_m in a mobile client m's surrounding cell, exceeding of which would result in a work mode shifting from the Direct mode to the P2P mode.

The AS first chooses the base cell size s_0 and the space is gridded into $n \times n$ base cells of size s_0. The AS decides s_0 based on the number of mobile clients and possible server work load(The role of s_0 will be clear after the description of service request algorithm of the mobile clients). Note that in the original centralized scheme, s_0 determines the best service quality because it is the smallest region that AS can provide. In the *HiSC*, however, with the complementary service quality of the P2P mode, the service quality is no longer restricted by s_0.

Each mobile client needs to register on the AS before its first LBS request attempt. Through the registration, each client m learns s_0 from the AS. Then m chooses s_0, or any ancestors of s_0 in the quad tree structure in the AS, as its surrounding cell size s_m. In the next stage, m receives the initial working mode w_m from the AS. A mobile client m's working mode is determined using the following formula:

$$w_m = \begin{cases} 1(Centralized) & n_{s_m} < K_m \\ 2(Direct) & K_m \leq n_{s_m} < K_m + \epsilon_m \\ 3(P2P) & n_{s_m} \geq K_m + \epsilon_m \end{cases}$$

where K_m is the $\mathcal{K}$-anonymity requirement of m, n_{s_m} is the number of clients in m's surrounding cell s_m, and ϵ_m is the tolerance count threshold.

4.1 *LBS* Request from Mobile Client m

To address the $\mathcal{K}$-anonymity requirement, each mobile client m chooses its own privacy level K_m. Each client also chooses a tolerance count ϵ_m and surrounding cell size s_m. In Figure 3, $s_0 = 1$, all the clients have $K_m = 5, \epsilon_m = 2$ and $s_m = s_0 = 1$(surrounding cell size is equal to base cell size).

Algorithm 1 Mode Maintenance for Mobile Client m

```
1: if (m reaches a new base cell i from base cell j ) then
2:    notify AS that m has moved from base cell j to base cell i;
3: end if
4: if (m receives new mode information wNew from AS) then
5:    w_m ← wNew;
6: end if
```

At any time, each mobile client m knows its working mode w_m. w_m is updated through client-server communication. m keeps track of its own location through location enabled devices such as GPS, and notifies AS only when it moves across a base cell. If m also leaves its own surrounding cell, it will receive a new w_m for the new surrounding cell. The new surrounding cell is a neighboring cell in the quad tree that m just entered, with the same size as the original surrounding

cell. In addition, m may also be notified about its new working mode when other peers move in and out of m's surrounding cell. As shown in Figure 3 (a) and (b), when client m_1 (yellow dot in color print) moves from cell j to cell i (from time t to $t+1$), a count update message will be sent to the AS to ask the AS to update the counts of cell i and cell j. Since m_1 also leaves its own surrounding cell, its working mode w_{m_1} will get updated by the AS from 1(*centralized*) to 3($P2P$). This is because $n_j = 2 < K_m = 5$ before m_1 leaves cell j at time t, and $n_i = 8 \geq K_m + \epsilon_m = 5 + 2 = 7$, after m_1 enters cell i at time $t+1$, assuming $\epsilon_m = 2$ and n_i, n_j represents the number of mobile clients in cell i and cell j respectively. The count maintenance algorithm is described in Algorithm 1.

Algorithm 2 *LBS* Request from Mobile Client m

```
1: if (w_m = Direct) then
2:    use m's surrounding cell as the cloaking region to request LBS
3: else if (w_m = P2P) then
4:    call RRS(Random Range Shifting) algorithm to calculate a random cloaking
      region c_hmax, so that all clients in c_hmax are within hmax hop distance of m
5:    if (n_{c_hmax} ≥ K_m) then
6:       //n_{c_hmax} is the number of mobile clients in region c_hmax
7:       request LBS using c_hmax
8:    else
9:       ask the AS to calculate the cloaking region R and request LBS
10:      get the result from the AS and filter the accurate answer(s)
11:   end if
12: else
13:   //w_m = Centralized
14:   ask the AS to calculate the cloaking region R and request LBS;
15:   get the result from the AS and filter the accurate answer(s)
16: end if
```

Now we discuss the mechanism of requesting *LBS* from a mobile client and illustrate the relationships among *HiSC*, centralized schemes, and P2P based schemes afterward. We will show that by tuning the two parameters, *HiSC* degenerates into one of the approaches proposed in the literature.

When a mobile client m initiates a *LBS* request, if its surrounding cell already has no less than K_m but less than $K_m + \epsilon_m$ mobile clients including itself, i.e. $K_m \leq n_{s_m} < K_m + \epsilon_m$, m's surrounding cell s_m will be used as the cloaking region. If its surrounding cell has no less than $K_m + \epsilon_m$ mobile clients, m decides to initiate a peer searching process based on the P2P scheme. If this process finds at least $K_m - 1$ peers, the region calculated by the P2P based scheme will be used as the cloaking region.

In practice, the P2P search process tends to find a region with m at the center of it, leading to possible easy association of a service request to a particular mobile client at the center of the cloaking region if location information is obtained by the adversary[14]. Therefore we propose a RRS(random range

shifting) algorithm (which will be explained shortly after) to avoid the possible privacy leakage.

Due to communication range constraint, even though there are more than K_m mobile clients around m, the P2P search process may not find enough peers. For example, mobile client m in Figure 2 (b) would not be able to find enough peers for $K_m = 5$. We propose the following remedy scheme. If less than $K_m - 1$ peers are found during the process, m will ask AS to calculate the cloaking spatial region.

In Figure 3, the number of mobile clients in m_2's (red dot in color print) surrounding cell is 1 and is less than $K_m = 5$, therefore m_2 asks the AS to calculate a larger region(a higher level in the quad-tree). This region (which has a count value of 5) will be used for service request. On the other hand, the number of mobile clients in m_3's (blue dot in color print) surrounding cell has a count value of 8. If $\epsilon_m >= 3$, m_3 will issue the service request directly using cell h. Otherwise, m_3 will apply a P2P based approach to find the dark shaped region (blue in color print) as the cloaking region.

4.2 RRS (Random Range Shifting) Algorithm

Algorithm 3 Random Range Shifting for Mobile Client m

1: $(m_x, m_y) \leftarrow m$'s coordinate
2: find collection of m's neighbors $N(m) = \{p | p$ is single-hop reachable from $m\}$
3: find the smallest circular region c that covers $N(m)$ with c center (c_x, c_y) and radius r
4: generate two random numbers δ_x and δ_y so that $m_x - c_x - \frac{r}{\sqrt{2}} \leq \delta_x \leq m_x - c_x + \frac{r}{\sqrt{2}}$ and $m_y - c_y - \frac{r}{\sqrt{2}} \leq \delta_y \leq m_y - c_y + \frac{r}{\sqrt{2}}$
5: shift c to obtain c' so that the center of c' is $(c_x + \delta_x, c_y + \delta_y)$ while radius r is unchanged
6: find subset $N_0(m)$ of $N(m)$, where $N_0(m) = \{p | p \in N(m) \wedge p$ is not inside $c'\}$
7: suppress all clients in $N_0(m)$ during the P2P communication
8: $c_h \leftarrow c'$
9: $h \leftarrow 2$, $hmax \leftarrow$ Max Hop Distance
10: **while** ($n_{c_h} < K_m$ && $h \leq hmax$) **do**
11: use h-hop P2P communication to find cloaking region c_h
12: $c_{hmax} \leftarrow c_h$
13: $h \leftarrow h + 1$
14: **end while**

As algorithm 3 shows, m starts the peer searching with finding its single-hop neighbor set $N(m)$. If we try to find the smallest region that contains $N(m)$ and clients are random distributed, m tends to be the center of this initial region. Instead, we go one step further, i.e., the initial region c is random shifted to be c' so that m may be in any position inside c'. The new c' contains less clients than original c. Then we continue the peer searching process using a subset of

$N(m)$ contained in the random shifted region, while suppress the rest part of $N(m)$ until we find K_m clients or reach the maximum hop distance.

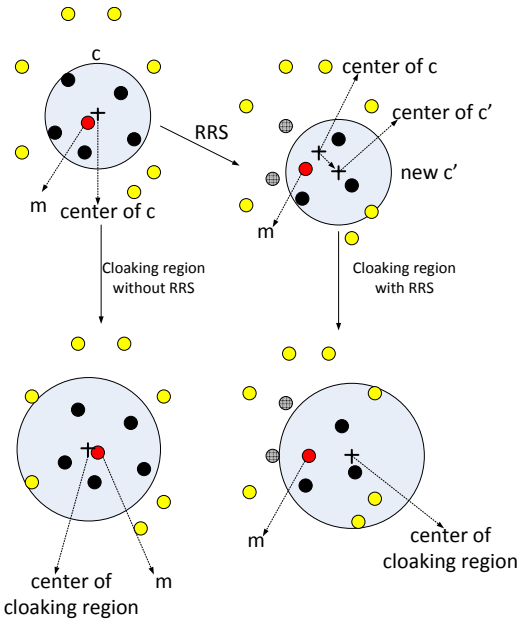


Fig. 4. RRS Algorithm Illustration ($K_m = 7$ for mobile client m)

Figure 4 illustrates the process of *RRS* algorithm. m is the red dot in color print. Assuming $K_m = 7$, originally m is near the center of the cloaking region. However, if c is shifted, m is no longer close to the center of the new cloaking region.

We can prove that after the shift, m is still contained in the new shifted region. This is because the new region's center is $(c_x + \delta_x, c_y + \delta_y)$, the distance between m and the center is $\sqrt{(m_x - c_x - \delta_x)^2 + (m_y - c_y - \delta_y)^2}$. According to the condition of δ_x and δ_y, this distance is not bigger than radius r. On the other hand, δ_x and δ_y are random values, therefore m can be at any location inside the new c'.

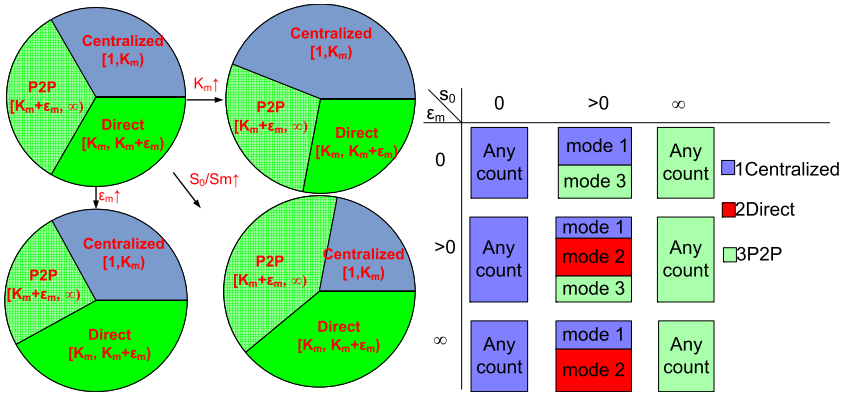
4.3 AS Algorithm

The *AS* maintains the count of each cell in every level of the quad tree structure. The quad tree is updated all the way through to the top level whenever a client moves across the boundary of a base cell. Relevant mobile clients will be notified if their working modes change.

When the AS receives a service request with an anonymity level of K_m from a mobile client m, the AS will find the smallest cell in the quad tree structure that contains at least K_m clients including m.

Algorithm 4 AS Algorithm

1: **if** (the AS receives that m reaches a base cell i from a base cell j) **then**
2: $count(i) + +$;
3: $count(j) - -$;
4: notify each mobile client its new mode wherever there is a mode change due to m's moving into and out of its surrounding cell;
5: **end if**
6: **if** (the AS receives a LBS request from m) **then**
7: Let i_m be the base cell that contains m;
8: $R \leftarrow i_m$;
9: **while** ($count(R.parent) < k$) AND $R!$ = the whole area) **do**
10: $R \leftarrow R.parent$;
11: **end while**
12: the AS uses R as the cloaking region to request LBS and filter results for m;
13: **end if**



(a) Workload Distribution (b) Load Balancing Using s_m and ϵ_m

Fig. 5. Hybrid Scheme Workload

4.4 The Relationships among *HiSC*, Centralized Scheme, and P2P based Scheme

We give insights on the behavior of the $HiSC$ and its degeneration into the P2P scheme or the centralized scheme under specific parameter setups. Whenever a

mobile client m requests *LBS*, if less than K_m clients exist in m's surrounding cell, m will request the *AS* to calculate a region by using the quad tree. This works the same way as the centralized scheme, except that there are much less location updates from mobile clients to the *AS*, and *HiSC* does not need a high resolution base cell to keep the service quality. As shown in Figure 5 (a), for fixed overall K_m and ϵ_m, when the base cell size s_0 or the surrounding cell size of a mobile client s_m increases, it is more likely to have enough peers in the surrounding cell of m. Therefore the centralized mode will take over less load using the proposed algorithm. Thus s_0 and s_m can be used to balance the work load between the *AS* and the mobile clients.

When the count in m's surrounding cell ranges from K_m to $K_m + \epsilon_m$, m will request service directly using its surrounding cell. If more than $K_m + \epsilon_m$ clients exist in m's surrounding cell, m will issue a P2P communication cycle to find at least $K_m - 1$ peers and calculate a cloaking region to request the service. This works similarly as the P2P based scheme except that service can always be fulfilled regardless of number of clients through a remedy request to the *AS*. As shown in Figure 5 (a), with other parameters fixed, when ϵ_m increases, the work load allocated to P2P scheme is shifting toward direct mode. Thus ϵ_m can be used to balance the work load between the Direct mode and P2P mode.

The direct mode has least latency because neither P2P searching nor server process queue waiting is involved. If response time is the only concern, we would increase ϵ_m to encourage direct mode. However, this in general will degrade service quality in terms of larger cloaking region.

When K_m increases, m is less likely to have enough peers in its surrounding cell. As a result, more requests are sent to *AS*, increasing the server work load.

Figure 5 (b) shows that when s_0 is zero, mobile client m will constantly update *AS* about its location because it moves across base cell boundary whenever it moves. All the service requests will be sent to the *AS* because the surrounding cell s_m (essentially also size 0) does not contain any peers. *HiSC* degenerates into the centralized scheme. When s_0 is the whole space, mobile clients do not update *AS* about their location because they never move out of a base cell (essentially the whole space). All the service requests will be satisfied with P2P based scheme because there are always more than enough peers in the whole space (assuming there are at least K_m users). In this case, *HiSC* degenerates into the P2P based scheme. In general, if ϵ_m is zero, there is no direct mode except when n_{s_m} is exactly K_m. When ϵ_m is infinite, there is no P2P mode.

Note that there are significant differences between our *HiSC* approach and original centralized or P2P based scheme. Instead of periodically updating the location information, a client only report the count update("count update" means an increase("+") to the count of the cell it moves into and a decrease("-") to the count of the cell it leaves) when it moves across base cell boundaries. With complementary service quality provided by P2P mode, the *AS* does not need a high resolution base cell. Therefore the latency caused by large number of location updates and service requests can be substantially alleviated. On the other hand,

keeping the AS in the system would provide consistent service regardless of client distributions, which a pure P2P approach cannot achieve independently.

5 Algebraic Cost Models for System Evaluation

For any privacy-preserving location cloaking system, a fundamental metric is how well the query anonymity is preserved. Any query should be fully cloaked in a $\mathcal{K}$-anonymity set, i.e., no privacy leakage is allowed. Beyond that, the system would preferably achieve optimal performance in terms of service quality, response time, and message cost. Previous sections have addressed the issue of how the $\mathcal{K}$-anonymity is effectively maintained in the proposed *HiSC* approach. This section will mainly focus on algebraic performance metrics as well as providing some tuning guidelines.

5.1 Work Load Distribution

One of the aims in designing *HiSC* is to balance the work load between the AS and the mobile clients. The actual distribution of the work load depends on user-defined parameters (such as K_m, s_m, and ϵ_m), number of clients, and locations/trajectories of the clients. Let p_m^1, p_m^2, and p_m^3 denote the average probability of a client m using cloaking method 1(Centralized), 2(Direct), and 3(P2P) respectively, straightforwardly we have:

$$p_m^1 = p_{\{w_m=1\}} = p_{\{n_{s_m} < K_m\}} \tag{1}$$

$$p_m^2 = p_{\{w_m=2\}} = p_{\{K_m \leq n_{s_m} < K_m+\epsilon_m\}} \tag{2}$$

$$p_m^3 = p_{\{w_m=3\}} = p_{\{(n_{s_m} \geq K_m+\epsilon_m) \wedge (n_{c_{hmax}} \geq K_m)\}} \tag{3}$$

Note that the equation for p_m^3 does not consider the case when P2P mode fails to fulfill m's service request. Additionally we use p_m^4 to represent the average probability that P2P mode fails. We have

$$p_m^4 = 1 - p_m^1 - p_m^2 - p_m^3 \tag{4}$$

In such case, the request will be redirected to the AS.

At any time, n_{s_m} is determined by the number of clients, the spatial distribution of the clients, base cell size s_0, and surrounding cell size s_m. The number and distribution of the clients can be statistically measured. For any given client distribution and K_m, the increase of s_0(globally) or s_m(individually) would cause n_{s_m} to increase, resulting less percentage of work load on the AS. In other words, $(s_0, s_m) \uparrow \Rightarrow p_m^1 \downarrow$. The increase of ϵ_m, on the other hand, will result in higher percentage of work load in direct mode, i.e., $\epsilon_m \uparrow \Rightarrow p_m^2 \uparrow$.

It can also be observed from the above equations that when K_m decrease or number of clients increase, the AS will also enjoy less percentage of work load. In addition, a P2P search that leads to a centralized mode due to the inability to find enough neighbors, i.e., p_m^4 is highly dependent on client distribution.

5.2 Service Quality

In location based services, it is desirable to provide most accurate answers to the users. This is often not possible in the context of preserving $\mathcal{K}$-anonymity for the requesters. In Section 6, we will use number of candidates that are the actual answers to measure the service quality in real queries. In theory, it is sufficient to use the size of the cloaking region to measure the service quality because it directly determines the size of the candidate list.

In general, a P2P based scheme can calculate a more precise cloaking region because of the inherent peer to peer communication mechanism, although sometimes it cannot find such a region to fulfill the request. The centralized scheme, on the other hand, cannot always guarantee a small cloaking region because when the *AS* recursively go through the quad tree to the upper level, the resolution of the cell decrease substantially. Intuitively the average size of cloaking regions found by *HiSC* approach would fall in the middle of the two approaches. In the real query examples shown in our experiments, however, *HiSC* provides results that are almost as accurate as the P2P scheme.

Formally, for any client m, the service quality q_m measured as the cloaking region size can be estimated using following formula[2]:

$$q_m = \begin{cases} q_m^1 = 4^k \times s_0, k = 0, 1, 2... & w_m = 1 \\ q_m^2 = s_m & w_m = 2 \\ q_m^3 = c_{hmax} & w_m = 3 \\ q_m^4 = 4^k \times s_0 & w_m = 3 \end{cases}$$

where k represents quad tree level on the *AS*, and c_{hmax} is the output of the random range shifting algorithm. The average service quality is therefore

$$\overline{q_m} = \sum_{i=1}^{4} p_m^i \times q_m^i \tag{5}$$

To achieve better service quality, we need to reduce $\overline{q_m}$. Note that in general $q_m^3 < q_m^2 < q_m^1$(because p2p mode can provide the smallest cloaking region, and centralized mode has the largest cloaking region), we may reduce p_m^1 or increase p_m^3 using the guidelines in the previous subsection, i.e., increase s_0 and/or decrease ϵ_m.

5.3 Response Time

Applications in the *LBS* often require near real time responses. In practice, query results should return to users within a few seconds. The *Response Time* is defined as the elapsed time from the moment that a client m issues a request to the time that m receives query results. Since sending m's requests to the *AS*/*LBS* and *AS*/*LBS*'s response backs to m require fixed time (and are almost instantaneous),

[2] The formula is only an estimation and used for tuning guidelines. Due to random range shifting and other factors, the actual size could be different.

we will use spatial cloaking time(including waiting time) to evaluate the response time for client m, which we denote as t_m.

In a centralized scheme, t_m is largely determined by the number of client updates and requests. For light workload, the AS can find the cloaking region for any client very quickly. However, when there are large number of clients, m may have to wait in the AS's task queue until location updates and other clients' requests have been fulfilled, resulting in large latency. In a P2P based scheme, t_m is mainly the time elapsed in the peer searching process. Large latency may occur when there are not enough peers near m. In general, we may estimate t_m using the following formula:

$$t_m = \begin{cases} t_m^1 \approx (n_u + n_r) \times t_{AS} & w_m = 1 \\ t_m^2 = 0 & w_m = 2 \\ t_m^3 \approx 2 \times K_m \times t_{P2P} & w_m = 3 \\ t_m^4 \approx 2 \times K_m \times t_{P2P} + (n_u + n_r) \times t_{AS} & w_m = 3 \end{cases}$$

where n_u and n_r are number of updates and number of requests that the AS needs to process at the time when m issues the request, t_{AS} represents the AS's unit processing time, and t_{P2P} denote the unit time of P2P communication. Note that when P2P mode fails to find enough peers, the request will be redirected to the AS, requiring additional time in getting response from the AS.

The average spatial cloaking time is

$$\overline{t_m} = \sum_{i=1}^{4} p_m^i \times t_m^i \tag{6}$$

To reduce response time, we may either try to increase p_m^2 (increase ϵ_m), or try to reduce number of update messages n_u using the proposed *HiSC*'s lazy update mechanism.

5.4 Message Cost

The *HiSC* generates three type of messages: *LBS* request message, location update message and peer searching message. By using lazy update mechanism, the *HiSC* has less update messages than the original centralized scheme. By limiting value of p_m^3(percentage of P2P mode), the *HiSC* also enjoys much less peer searching messages.

5.5 Tuning Guidelines

We summarize above cost models and provide following tuning guidelines. By tuning $s_0(s_m)$ and ϵ_m, a user can trade off response time and service quality using these guidelines.

Parameter	service quality q_m (cloaking region size)	Response Time t_m	AS's work load percentage p_m^1
$s_0(s_m)$	$s_0(s_m) \uparrow \Rightarrow q_m \downarrow$	$s_0(s_m) \downarrow \Rightarrow t_m \downarrow$	$s_0 \uparrow \Rightarrow p_m^1 \downarrow$
ϵ_m	$\epsilon_m \downarrow \Rightarrow q_m \downarrow$	$\epsilon_m \uparrow \Rightarrow q_m \downarrow$	N/A

Table 1. Performance Tuning Guidelines

6 Experiment

In this section, we evaluate the performance of the proposed *HiSC* approach and validate the algebraic cost models in the previous section. The results of the *HiSC* are compared with those of original centralized and P2P based schemes.

6.1 Experiment Settings

The spatial space is divided into multiple cells. The AS maintains the cells using a quad tree structure. The AS can determine the base cell size s_0 based on the number and average speed of registered clients, response time and service quality requirements, and its work load. Unless specified, in all the experiments, we use the *Network based Generator of Moving Objects*[24] to generate moving objects using a real road network. The real query examples(K nearest neighbor query) are used in the experiments. The target objects are uniformly distributed in the space. The initial locations and moving trajectories of all the clients are constrained by the road network.

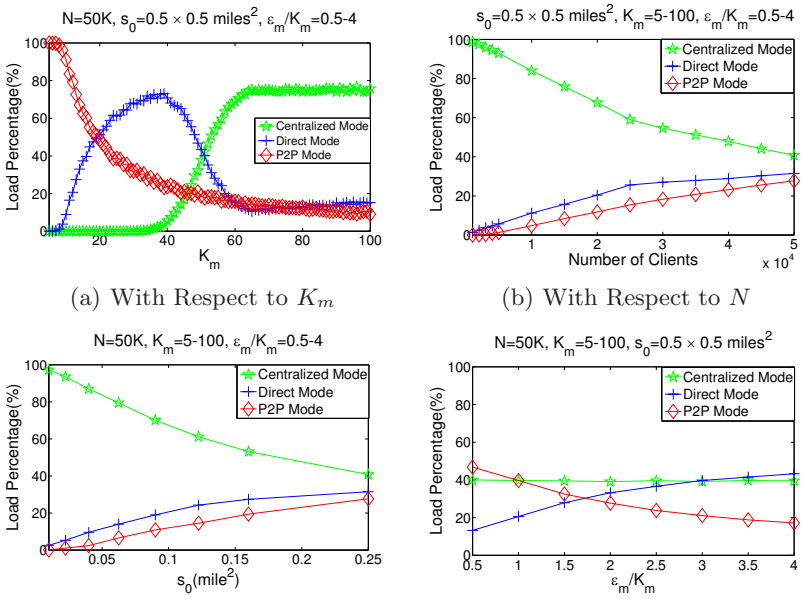
Unless specified, our simulation is running over 50K clients and 3K static query targets. Each client m can use its own GPS device to get updated location information periodically. m may specify a privacy parameter K_m ranging in $[5, 100]$, a tolerance count ϵ_m, and a surrounding cell size s_m that is equal to the size of base cell s_0 or any ancestor of base cell in the AS's quad tree structure.

Whenever m's location information is updated, m may change its speed in the next location update cycle. m's speed falls in the range of $[0, 60]$ miles per hour. m's probability of sending LBS requests follows an exponential distribution. We further assume that m can communicate with other clients through a $1Mbps$ wireless LAN channel. m can communicate with the AS or LBS provider through the base station with $10Mbps$ bandwidth. When m sends a request to the AS, it has to wait in the AS's process queue if the AS is processing location updates or requests from other clients. In order to send a peer search request, m also needs to wait until the wireless channel is not busy. For measurement simplicity, we consider each message sent or received has the size of $1KB$, and the AS's processing time unit is 0.1ms.

Note that changing s_m(in the client level) would exhibit similar effects as changing s_0(in the system level), we only show the figures of latter in the following experiments.

6.2 Work Load Distribution

Our first experiment is to evaluate how the work load distribution can be affected by K_m and number of clients N. We will also show how the tuning of s_0(or s_m) and ϵ_m can degenerate the *HiSC* into three different working modes, namely, Centralized mode, Direct mode, and P2P mode.



(a) With Respect to K_m (b) With Respect to N

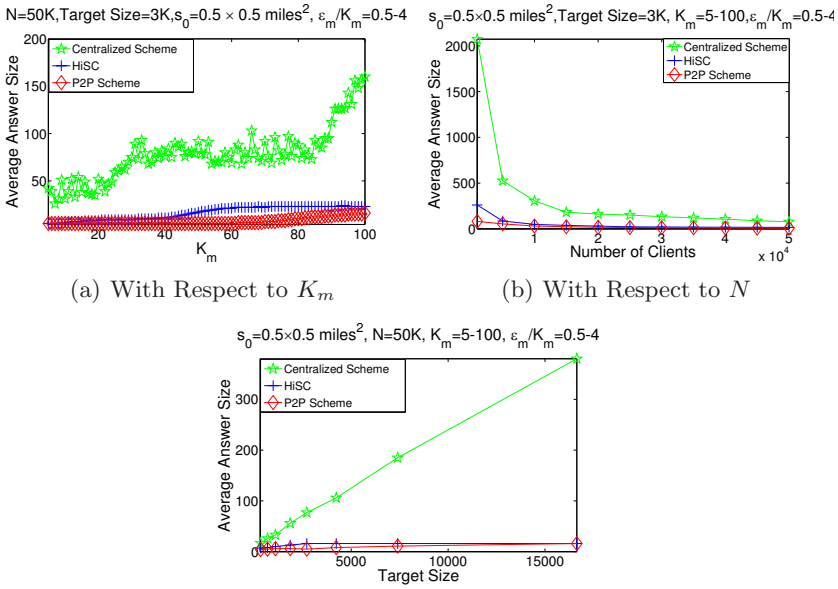
(c) With Respect to s_0 (d) With Respect to ϵ_m

Fig. 6. Work Load Distribution

As shown in Figure 6 (a) and (b), the work load distribution is affected by K_m and number of clients N. As expected, when K_m increases or N decreases, *HiSC* tends to reduce *AS*'s work load percentage. Figure 6 (c) indicates that we may increase the size of s_0 to reduce the percentage of Centralized work mode, thus reducing the *AS*'s work load. We may also increase the value of tolerance count ϵ_m, as Figure 6 (d) shows, to give preference to the Direct mode.

6.3 Service Quality

The next experiment is to compare the service quality of the *HiSC* approach with original centralized and P2P based scheme. In our experiments, we use real query examples(k nearest neighbor query where $k = 4$) to measure the size of answers, which is directly related to the size of the cloaking region. When K_m



(a) With Respect to K_m (b) With Respect to N

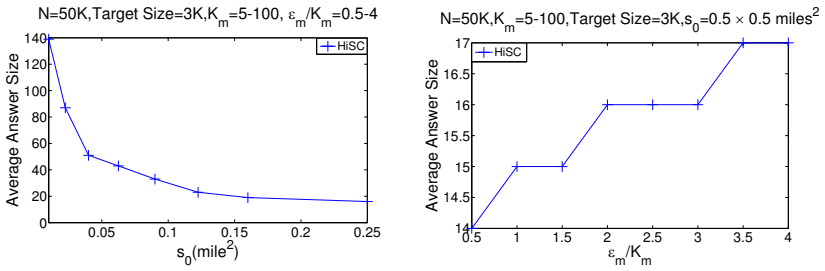
(c) With Respect to Target Size

Fig. 7. Answer Size Comparison

increases or number of clients decreases, larger cloaking region size is required, resulting degrade of service quality, as can be observed from Figure 7 (a) and (b). In general, the *HiSC* approach's service quality is close to that of the P2P based scheme especially when K_m is not very large, or when there are large number of clients. This is because *HiSC* tends to prefer P2P or Direct working mode in these cases. The scalability of the *HiSC* is also exhibited in Figure 7 (c), which shows that when query target size increases, the *HiSC* provides results that are almost as accurate as the P2P scheme. In all the experiments, the *HiSC* provides much better service quality than the original centralized scheme given same base cell size s_0. Figure 8 shows how the tuning of s_0 and ϵ_m can contribute to the service quality. By increasing s_0 and/or decreasing ϵ_m, the *HiSC* can produce more accurate answers without losing $\mathcal{K}$- anonymity. However, the role of ϵ_m is less significant in tuning service quality than in response time, as demonstrated in the next subsection.

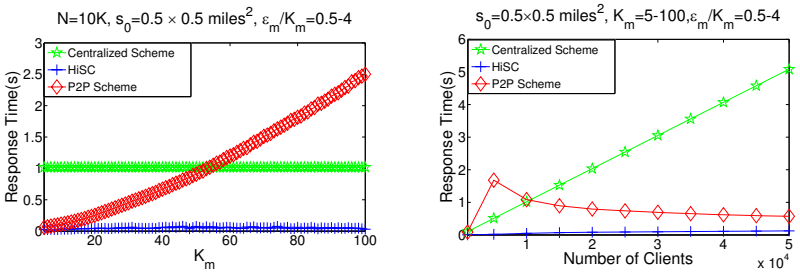
6.4 Response Time

We use spatial cloaking time(including waiting time) in our experiments to evaluate the response time. Figure 9 (a) indicates that in a P2P based scheme, the response time increases linearly with K_m, while the centralized scheme and the *HiSC* are less sensitive to the change of K_m. In Figure 9 (b), the response time



(a) With Respect to s_0 (b) With Respect to ϵ_m

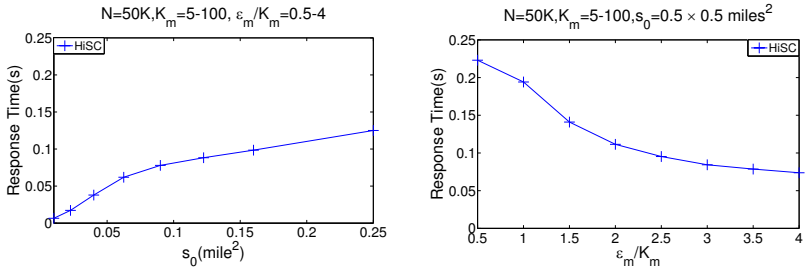
Fig. 8. Answer Size Tuning



(a) With Respect to K_m (b) With Respect to N

Fig. 9. Response Time

for the centralized scheme increases slightly superlinearly with number of clients, while the P2P scheme and the *HiSC* are more scalable. This is because that, in the original centralized scheme, the AS needs to handle lots of location updates and requests at the same time, resulting longer average waiting time when there are large number of clients. By contrast, the *HiSC* enjoys less location update, and can switch to Direct mode or P2P mode to reduce response time. As shown in both figures, the *HiSC* can reduce the response time by more than an order of magnitude from both the P2P scheme and the centralized scheme when K_m is large($K_m > 50$) or number of clients is large($N > 10K$).



(a) With Respect to s_0 (b) With Respect to ϵ_m

Fig. 10. Response Time Tuning

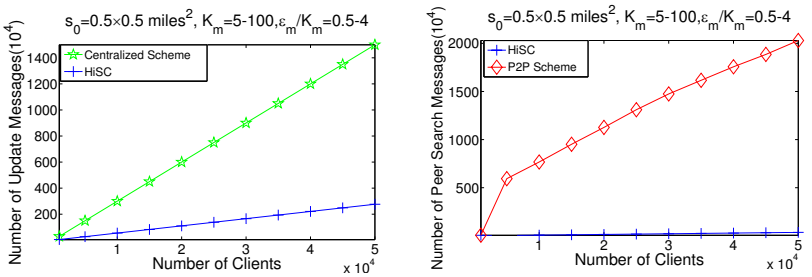
By tuning s_0 (decrease) and/or ϵ_m (increase), the *HiSC* can achieve better response time, as shown in Figure 10 (a) and (b). Combining the results with that of previous subsection, we may find that by appropriate selection of s_0(or s_m) and ϵ_m, the *HiSC* can enjoy optimized response time and service quality.

6.5 Message Cost

An alternative way to evaluate work load is to calculate the message cost in the system. Since request messages are not bound to any approach, our experiments will focus on location update messages(for *HiSC* and centralized scheme) and peer searching messages(for *HiSC* and P2P based scheme). As shown by Figure 11 (a) and (b), the *HiSC* can reduce the update message cost of the AS by almost 6 times and the peer searching message cost of the clients by more than an order of magnitude.

6.6 Effectiveness of RRS algorithm

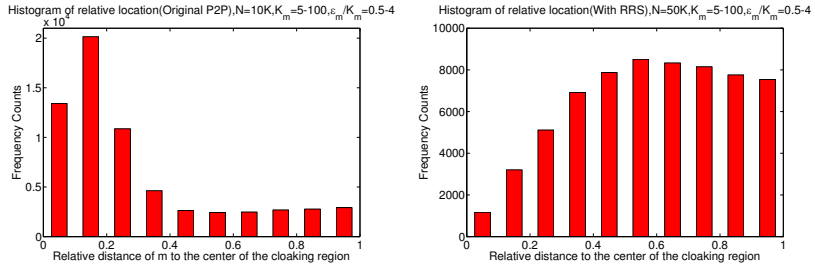
Now we evaluate the effectiveness of our proposed RRS(random range shifting) algorithm in protecting privacy leakage. To measure the relative location of each client $m(m_x, m_y)$ to the center of the cloaking region c, the distance from m to



(a) With Respect to N (b) With Respect to N

Fig. 11. Message Cost

the center is mapped into a double-precision number $\hat{d}$ ranging in $[0, 1]$, where 0 means that m is at the center, and 1 means m is at the boundary of c. The statistics of $\hat{d}$'s distribution can then be collected. As can be seen from Figure



(a) Without RRS (b) With RRS

Fig. 12. Histogram of client's relative location

12(a), without RRS, m tends to be close to the center of the cloaking region. While in Figure 12(b), m is almost evenly distributed in the cloaking region. (Note that when distance increases, m's possible location spans a larger area, therefore ideally the sample size should also increase). This proves that our RRS algorithm successfully avoid the possible privacy leakage caused by m's large possibility at the center of the cloaking region.

6.7 Service Fulfillment Rate

Finally, we evaluate the service fulfill rate in the original P2P based scheme. As shown in Table 2, when number of clients is not large enough, the original P2P

N=1K	N=2K	N=3K	N=4K	N=5K
1.4%	21.4%	85.0%	97.6%	99.5%

Table 2. Service Fullfill Rate with Respect to Number of Clients(Original P2P)

scheme may has low service fulfill rate. On the other hand, our proposed *HiSC* can always guarantee the service to be fulfilled regardless of client distribution.

7 Conclusion and Future Work

In general, our proposed *HiSC* approach effectively preserves query anonymity. It provides an effective way to balance the work load among the *AS* and the mobile clients. By integrating salient features of the centralized scheme and P2P based scheme, it can provide high service quality without maintaining high-resolution counts information at the *AS* side. The new lazy update mechanism in the *HiSC* allows the *AS* to enjoy much less work load caused by massive location updates from the clients. The *HiSC* also provides more scalable and consistent service. It guarantees service availability even when clients are sparsely distributed. By tuning one system parameter (*base cell size*) and three client parameters(*anonymity level*, *surrounding cell size* and *tolerance count*), the *HiSC* can provide personalized privacy, response time and service quality requirements. Additionally, we designed a simple yet effective RRS(random range shifting) algorithm to prevent possible privacy leakage that would exist in the original P2P approach. As our experiments shows, the *HiSC* provides close to optimal service quality, while greatly reduces the response time and message cost.

In our future work, we plan to build a prototype system that consists of all components in our approach. We are also looking into the problems of locational privacy beyond $\mathcal{K}$-anonymity.

References

1. Snekkenes, E.: Concepts for personal location privacy policies. In: EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce, New York, NY, USA, ACM (2001) 48–57
2. Langheinrich, M.: Privacy by design - principles of privacy-aware ubiquitous systems. In: UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing, London, UK, Springer-Verlag (2001) 273–291
3. Duri, S., Gruteser, M., Liu, X., Moskowitz, P., Perez, R., Singh, M., Tang, J.M.: Framework for security and privacy in automotive telematics. In: WMC '02: Proceedings of the 2nd international workshop on Mobile commerce, New York, NY, USA, ACM (2002) 25–32
4. Ardagna, C.A., Cremonini, M., Damiani, E., di Vimercati, S.D.C., Samarati, P.: Supporting location-based conditions in access control policies. In: ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security, New York, NY, USA, ACM (2006) 212–222

5. Zibuschka, J., Scherner, T., Fritsch, L., Rannenberg, K., Goethe, J.W.: Towards a unified interface for privacy regulation-conformant location-based services. In: W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement, Ispra/Italy (October 2006)
6. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10(5)** (2002) 557–570
7. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10(5)** (2002) 571–588
8. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services, New York, NY, USA, ACM (2003) 31–42
9. Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, Washington, DC, USA, IEEE Computer Society (2005) 620–629
10. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preserving anonymity in location based services. Technical report, National University of Singapore (2006)
11. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: VLDB '06: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment (2006) 763–774
12. Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), Washington, DC, USA, IEEE Computer Society (2005) 194–205
13. Cheng, R., Zhang, Y., Bertino, E., Prabhakar., S.: Preserving user location privacy in mobile data management infrastructures. In: PET '06: 6th Workshop on Privacy Enhancing Technologies. (2006)
14. Ghinita, G., Kalnis, P., Skiadopoulos, S.: Prive: anonymous location-based queries in distributed mobile systems. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 371–380
15. Ghinita, G., Kalnis, P., Skiadopoulos, S.: Mobihide: A mobile peer-to-peer system for anonymous location-based queries. In: SSTD '07: 10th International Symposium on Advances in Spatial and Temporal Databases, Boston, MA, USA, Springer (2007) 221–238
16. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Pervasive 05': Third International Conference on Pervasive Computing. (2005) 152–170
17. Schilit, B.N., LaMarca, A., Borriello, G., Griswold, W.G., McDonald, D., Lazowska, E., Balachandran, A., Hong, J., Iverson, V.: Challenge: ubiquitous location-aware computing and the "place lab" initiative. In: WMASH '03: Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots, New York, NY, USA, ACM Press (2003) 29–35
18. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems, New York, NY, USA, ACM (2006) 171–178

19. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: ICPS '05: Proceedings of IEEE International Conference on Pervasive Services. (July 2005) 88–97
20. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05), Washington, DC, USA, IEEE Computer Society (2005) 217–228
21. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), Washington, DC, USA, IEEE Computer Society (2006) 25
22. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2005) 49–60
23. Beresford, A.R., Stajano, F.: Mix zones: User privacy in location-aware services. In: Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. (Mar. 2004)
24. Brinkhoff, T.: A framework for generating network-based moving objects. Geoinformatica **6**(2) (2002) 153–180