國立交通大學

資訊科學與工程研究所

碩 士 論 文

以道路網路狀的遮蓋機制來保護位置隱私權

Spatial Network-based Cloaking Mechanisms for Location Privacy

研 究 生：李柏逸

指導教授：彭文志　教授

中 華 民 國 九十六 年 六 月

以道路網路狀的遮蓋機制來保護位置隱私權
Spatial Network-based Cloaking Mechanisms for Location Privacy

研 究 生：李柏逸　　Student：Po-Yi Li

指導教授：彭文志　　Advisor：Wen-Chih Peng

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

摘　　要

大部分的研究都致力於使用 *k*-匿名的方式來保護位置隱私權。要實做出 *k*-匿名最常見的架構是由一個可信任的伺服器(稱做位置匿名者)將至少 K 個使用者的位置做覆蓋，藉此來保護位置隱私權。而前人所做的研究都只有產生出矩形狀的覆蓋區域。然而，此矩形狀的覆蓋區域卻會導致較多的查尋結果，因而增加在過濾無用的解時的負擔。在這篇論文裡，我們提出了道路網路狀的覆蓋機制，可根據道路網路的特性來產生覆蓋區域。因為利用了道路網路的特性，此覆蓋區域將可以非常有效率的降低查詢結果的數量並且增加行動裝置的暫存資料使用率。另外，我們也提出了時間－空間相連性覆蓋的演算法(稱做 STCC)來建立道路網路的索引架構。我們也做了相關的模擬實驗。實驗結果顯示，我們所提出的 STCC 機制在候選查詢結果的數量以及暫存資料使用率的表現是較優勝的。

關鍵字：*k*-匿名，位置隱私權，位置匿名者。

Abstract

Most of research efforts have elaborated on k-anonymity for location privacy. The general architecture for implementing k-anonymity is that there is one trusted server (referred to as location anonymizer) that is responsible for cloaking at least K users' location for protecting location privacy. Prior works only generate grid shapes cloaking regions. However, grid shapes cloaking regions results in a considerable amount of query results, thereby increasing the overhead of filtering unwanted results. In this paper, we proposed spatial network-based cloaking mechanisms in which cloaking regions are generated according to the features of road networks. By exploring the features of spatial networks, the cloaking regions are very efficient for reducing query results and improving cache utilization of mobile devices. Furthermore, an index structure for spatial networks is built and in light of the proposed index structure, we develop *Spatial-Temporal Connective Cloaking mechanisms*(abbreviated as STCC). A simulation is implemented and extensive experiments are conducted. Experimental results show that our proposed mechanisms STCC outperforms prior cloaking algotihms in terms of the candidate query results and the cache utilization.

Keywords — k-anonymity, location privacy, location anonymizer

致　　謝

這兩年的碩士生活是一段紮實的學習過程。在這過程中我深深的體會到學習是沒辦法只靠我一個人的。首先，在論文上幫助我最多、最需要去感謝的人是我的指導教授彭文志老師。沒有老師帶著我一點一滴的前進，我只能在原地打轉找不到方向。因為每次在跟老師討論前，總是會有很多想不透的問題，經過與老師的討論後，都會有一種：哦～原來如此的感覺，讓我對我的論文越來越堅定。除了論文外，還有生活上的鼓勵，如同從小到大所遇過師長一般照顧我，因此我真得很感謝老師。其次，要很感謝我的口試委員陳良弼教授與黃俊龍教授，在口試中提供了我很寶貴的意見，讓我有機會可以去改善本篇論文中的缺失。

而在實驗室裡，首先最想要感謝的是博士班的學長洪智傑以及我們上一屆碩士班的學長李志劭、蕭向彥、楊慧友和張民憲。當我剛進實驗室還懵懂無知時，他們不吝惜的跟我分享學習的經驗，以及在學習過程中怎樣去面對困難，雖然他們跟我所做的領域不一樣，但他們卻都耐著性子陪我討論，我覺得他們真的很厲害。另外，還要感謝的是一直跟我奮戰到今天的碩士班同學游敦皓、黃正和以及周佳欣，大家一起修課、打球、吃飯、玩樂，更重要的是看到你們努力學習的樣子往往都會策發我要更加努力才行！最後還有碩士班的學弟蔡尚樺、駱嘉濠、蔣易杉和傅道揚，你們的資質真的都很棒，實力也很堅強，我好像沒什麼可以幫的上你們的地方，反倒是你們給了我許多的建議，讓我可以把論文修改的更好！

最後，要感謝的人是我的父母，我的父親在我碩二下的時候開始臥病在床，短短三個月便過世了，面對照顧父親與論文的煎熬，其實我一點辦法也沒有，但，我的父親到最後仍始終關心著我，我深深的體會到父親的愛。而我的母親背負著家裡的沈重，一天一天的消瘦，心裡以及身體的壓力可想而知，但她卻選擇讓我完成學業一個人苦苦的撐著，這是我母親的愛！得之於你們的愛實在是太多了，為了回報你們，因此我要更加努力學習才行！真的真的很謝謝你們！

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the advances in location detection devices (e.g., GPS devices, cell phones, RFIDs, etc.), mobile devices with computing, storage and wireless communication are increasingly popular recently. At the same time, map databases and geographical information services are widely used. Thus, a large number of location-based services (referred to as LBS) are now available and users could issue the location-based queries to the servers of LBS. Examples of location-based queries include *"when I am moving on a certain road, find the k nearest gas stations with me"* or *"what is the traffic condition within five minutes of my route"*. While LBSs have shown to be valuable to users' daily life, on the other hand, they also expose extraordinary threats to user privacy. If not well protected, the location information of users may be misused by some untrustworthy service providers or stolen by hackers. Once the location information is exposed, adversaries may dig for cues to invades user privacy. Obviously, it is important to protect location privacy.

Recently, the problem of location privacy preserving has received growing interests from the research community. Most of research efforts have elaborated on k-anonymity [5, 13, 21]. The general architecture for implementing k-anonymity is that there is one trusted server (referred to as location anonymizer) that is responsible for cloaking at least K users' location for protecting location privacy. Explicitly,

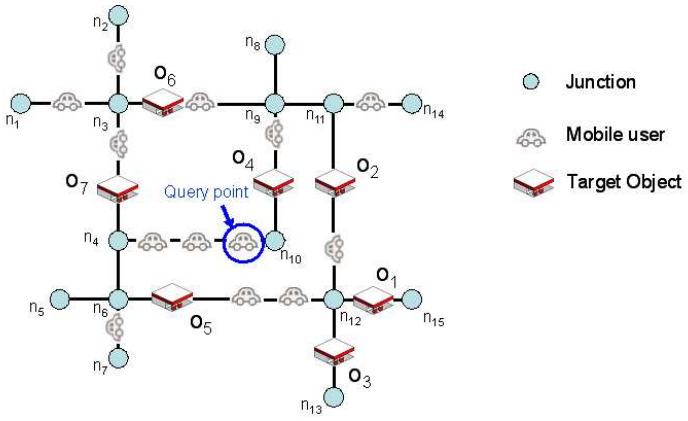
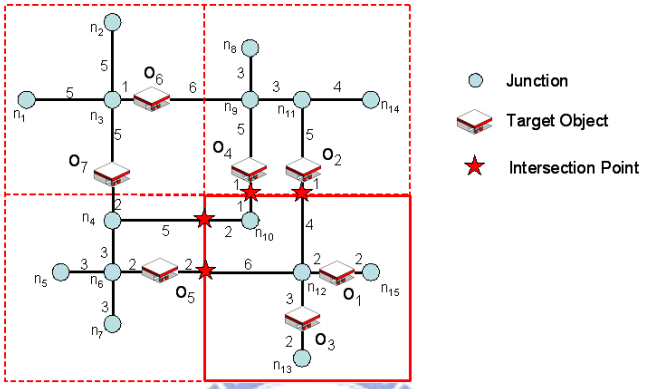
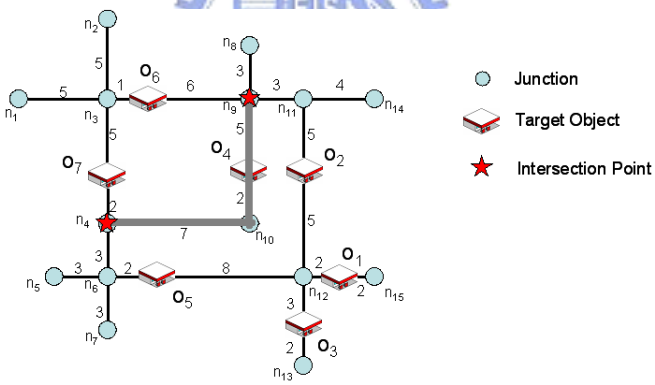


Figure 1.1: An example of spatial network.

a location anonymizer is built to collect user location and perform cloaking procedure in which the exact location of users is blurred as a cloaked spatial area in accordance with each user privacy requirements. Then, the location anonymizer will send location-dependent query alone with the cloaked spatial area to location-based server to retrieve location-dependent data. Note that since the query location is an area instead of single query point, location-dependent servers should fetch those query results based on the cloaked spatial area. Prior works in [14] proposed a framework for location services without compromising location privacy. However, only free space environment is considered, which is not realistic in a real world environment. Furthermore, the authors in [12] explore privacy protected query processing on spatial networks, where query processing schemes are modified to retrieve query results. Note that the cloaking mechanisms used in the prior works only consider cloaking regions are rectangle (i.e., grid-based shape). Only exploring grid-based shape for cloaking regions are not efficient due to that more candidate query results are retrieved. More candidate query results, more computing cost and communication cost. In addition, user's movements have spatial-temporal behavior, cloaking region based on grid-based shape can not efficiently improve cached utilization. Thus, in this paper, we intend to explore spatial network-based cloaking mechanism by taking features of spatial networks into consideration.

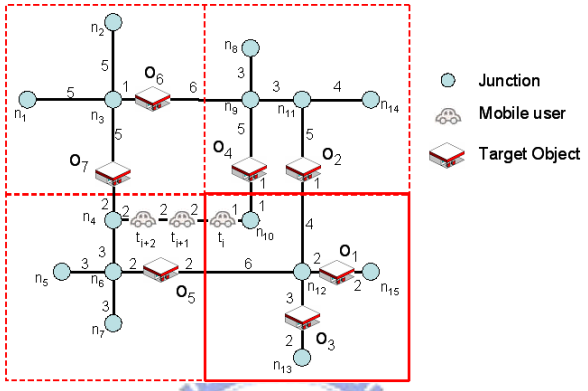


(a) 4-anonymity of grid-based mechanism

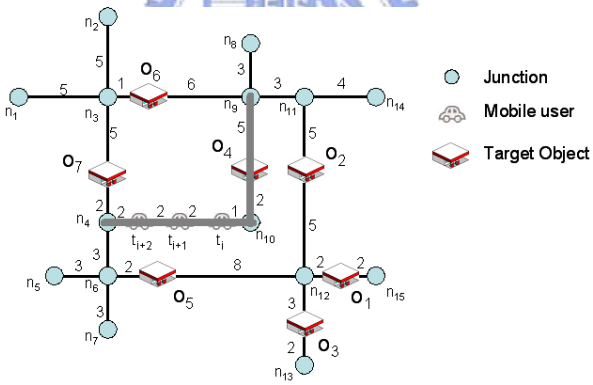


(b) 4-anonymity of spatial network-based mechanism

Figure 1.2: Examples of cloaking regions.



(a) 4-anonymity of grid-based mechanism



(b) 4-anonymity of spatial network-based mechanism

Figure 1.3: Examples of cache utilization.

The problem we study could be best understood by an illustrative example, where Figure 1.1 is the spatial network given. Figure 1.2 shows spatial network-based cloaking algorithm can retrieve less candidate query results than traditional grid-based cloaking mechanism. Figure 1.2(a) shows the cloaking regions of the traditional cloaking algorithm. It can be verified that the number of candidate query results is 5(i.e., O_1, O_2, O_3, O_4, O_5). On the other hand, Figure 1.2(b) shows the cloaked road segments of spatial network-based cloaking mechanism. Rather than blur query point into a cloaked region, we cloaked the query point into a series connective road segments(i.e. $(n_4, n_{10}), (n_9, n_{10})$). As shown in Figure 1.2(b), the number of candidate query results is 2(i.e., O_4 and O_7). Note that the traditional grid-based mechanism will obtain larger number of candidate query results is because grid-based mechanism averagely use more road segments to achieve k-anonymity. The more road segments in cloaked region, the more query result returned. For example, in Figure 1.2(a), traditional grid-based cloaking mechanism uses two users on segment (n_6, n_{12}), one user on segment (n_6, n_{12}) and query point to achieve 4-anonymity. There are 6 segments in the cloaked region. On the contrary, the spatial network-based mechanism, in Figure 1.2(b), only uses 2 road segments for cloaking. The spatial network-based cloaking mechanism can obtain less candidate query results and reduces the computing cost and communication cost. Note that cloaking regions based on spatial networks reflect spatial-temporal features. Thus, cache utilization of devices will be higher. Figure 1.3 shows the spatial network-based cloaking mechanism can efficiently reuse the cached data on mobile device. For each retrieving the candidate query results from LBS, the cloaked region and candidate query results' location will be temporary cached on mobile device. If a mobile user does not leave the cloaked region, he can find the query answer by filtering those candidate query results without issuing a query to LBS again. Consequently, increasing cache utilization is able to reduce the number

of queries, thus reducing revealed location information in issuing LBS queries. In Figure 1.3(a), a mobile user is moving from t_i to t_{i+2} and periodically issues query for LBS. The movement length of the mobile user in the cloaked region is the length summation of n_{10} to the two intersection point on segments (n_4, n_{10}) and (n_9, n_{10}). However, this mobile user cannot move to the other road segments in the cloaked region. Therefore, a mobile user will easily move out of the cloaked region on t_{t+1} and results in cache miss. The cloaked road segments of spatial network-based cloaking mechanism, in Figure 1.3(b), has the spatial-temporal connective property that fully fit the moving behaviors of users. Assume that at t_i, a mobile user retrieves the candidate query results(i.e., O_4 and O_7). Next, he calculates the network distance between his location to O_4 and O_7 and obtains O_4 is the query answer. Then, the cloaked road segments, $(n_4, n_{10}), (n_9, n_{10})$, and the location of query results will be cached on the mobile device. At next time slot, t_{i+1}, the cached data are still useful for this mobile user. It can be seen that by exploring spatial network features for cloaking, both candidate query size and cache hit are improved.

Consequently, in this paper, we propose spatial network-based cloaking mechanisms in which cloaking regions basically consist of road segments instead of grid-based shapes. Specifically, we propose a system consist of three components, mobile users, the *Spatial-Temporal Connective Location Anonymizer* and the *Privacy-Protected Query Processor*, to support privacy protected spatial queries for spatial networks. The mobile users can set their own privacy requirements by an user-specified privacy profile. The format of user-specified privacy profile is as (k, L_{min}) or (k, N_{min}) where k indicates the user wants to be k-anonymity, L_{min} indicates the user want the length of cloaked road segments is at least L_{min}, and N_{min} indicates the user want the number of cloaked road segments is at least N_{min}. Large value of k, L_{min} and N_{min} indicate more strict privacy requirements. The

Spatial-Temporal Connective Location Anonymizer can receives the exact location and privacy profile from mobile user. And the *Spatial-Temporal Connective Location Anonymizer* can blur mobile user's exact location into a series of connective cloaked road segments according to his privacy profile. For the *Spatial-Temporal Connective Location Anonymizer*, we propose a hierarchical index structure to decompose the spatial network into different level of granularity. It help us quickly find a series of connective road segments that best matches mobile user's requirement. The *Privacy-Protected Query Processor* can deal with privacy-protected query and returns the candidate query results instead of accurate answer according to the cloaked road segments. Finally, experimental results shows that our proposed cloaking mechanism fully utilize features of spatial networks. Thus, our proposal cloaking mechanisms are able to not only reduce candidate query results but also increase the utilization of caching query results.

The rest of the paper is organized as follows: Chapter 2 surveys the related work. Chapter 3 gives an outline of our system architecture. Chapter 4 and Chapter 5 respectively describe the main two components, the location anonymizer and the query processor, of our system architecture. Chapter 6 describes the experimental evaluation of our system. Chapter 7 concludes this paper.

Chapter 2

Related Works

The concept of k-anonymity [22] have been proposed and used to protect data privacy. For each data in database, the main idea is to let it is not distinguishable among other $k-1$ data to achieve k-anonymous. Gruteser et al. [5] used this concept and proposed spatial-temporal cloaking to protect mobile user's location privacy. They assume that all of the users have the same k-anonymous requirement. Once the user updates his location, the spatial space need recursively divided in a KD-tree-like format until the subspace is suitable for the k-anonymous requirement. But this method is not scalable because it needs run the *Adaptive-Interval Cloaking* algorithm again for each single movement of each user. Then Gedik et al. [4] proposed the *CliqueCloak* algorithm to support different k-anonymous requirement for each user. It constructs a clique graph and finds that some users can share the same cloaked region. Their minimum bounding rectangle is the cloaked region. However, these researches mainly focus on designing the location anonymizer rather than query processing. Molbel et al. [14] proposed a framework include three main components that are system user, location anonymizer and query processor. For mobile user, he can set his privacy profile to define his requirement like k-anonymous and A_{min} which is particular useful for dense area. Location anonymizer will construct a pyramid structure to index different granularity cloaked region.

Query processor will return candidate answers according to the cloaked region. However, both of the location cloaking and query processing are according to free space environment and not suitable for spatial network environments.

For the spatial network environment, we focus on the k-nearest-neighbor query. Kolahdouzan et al. [11] use a first order Voronoi diagram to efficiently evaluate the k-nearest-neighbor query on spatial networks. Papadias et al. [18] proposed the *Incremental Euclidean Restriction* algorithm (IER) and I*ncremental Network Expansion* algorithm (INE) to evaluate k-nearest-neighbor query on spatial networks. IER uses the Euclidean lower bound property. First, IER retrieves the Euclidean distance of query point to nearest neighbor as lower bound and calculate their network distance as upper bound. Next, if the next Euclidean nearest neighbor's network distance is smaller than before, updates the nearest neighbor. IER obtains the nearest neighbor by narrowing the search region until there is no other target object in this region. In addition, INE utilizes a priority queue to store the nodes and target objects to be explored through the expanding process. The nodes and target objects in the priority queue are sorted by their network distance to the query point. During the expansion, INE repeatedly dequeues the top entry in the priority queue and enqueues it's neighbor nodes and the target objects on the road segments to neighbor nodes with their network distance into the priority queue. When a target object entry is dequeued, the nearest neighbor is found. However, those spatial queries in the spatial network need accurate query point's location.

In order to provide privacy protected query process on spatial networks, Wei-Shinn Ku et al. [12] combine the concept in the work of [14] and [18]. They adopt the grid-based pyramid data structure proposed in [14] to provide cloaking function. Then their *Location-based Service Provider* has the ability to process the privacy protected query. They design *privacy protected spatial network nearest neighbor query* (PSNN) algorithm and *privacy protected spatial network range*

query (PSRQ) algorithm for two popular query types, nearest neighbor query and range query on spatial networks respectively. However, they focus on snapshot query. For continuous query, they will cause two problems: (1) the re-identified probability increasing problem and (2) the large costs of cloaking and query problem. Although they implement cache mechanism on mobile user's device, they still can not efficiently use the cache because of the character of traditional cloaked region described in the introduction section. Therefore, we need a new cloaked mechanism to efficiently process the continuous query on spatial networks.

Chapter 3

System Architecture

In this section, we describe the system architecture for supporting privacy protected spatial queries with underlay spatial networks. Figure 3.1 depicts our operating environment with three main entities: mobile users, the *Spatial-Temporal Connective Location Anonymizer*, and the *Privacy-Protected Query Processor*.

In spatial networks, they are usually modeled as an undirected graph $G(V, E)$, where V denotes the set of road junctions and E denotes the set of road segments. Mobile clients are distributed and move in it. We consider mobile clients such as cell phones, personal digital assistants (PDA), laptops, that are instrumented with a global positioning system (GPS) for continuous position information. Furthermore, we assume that there are access points/base stations around the system environment for mobile devices to communicate with the location cloaker. All users have

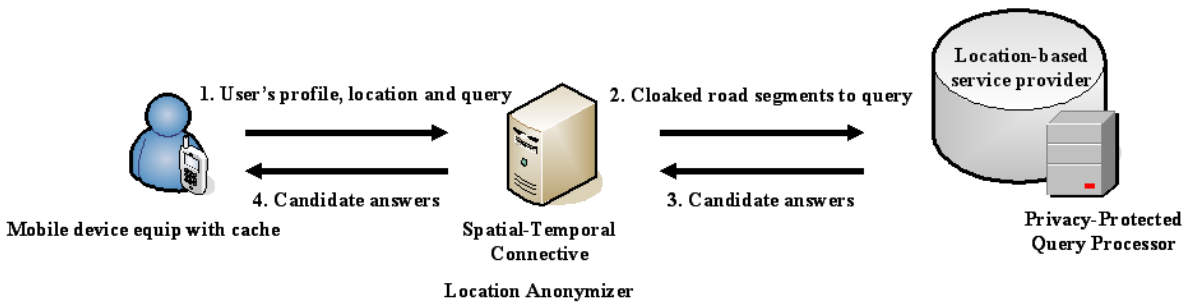


Figure 3.1: The system architecture.

mobility for traveling on underlay networks, continuously update their location to the *Spatial-Temporal Connective Location Anonymizer* and ask for LBSs. They also hold privacy policies which specify the privacy requirements of each user. A user privacy profile is defined as (k, L_{min}) or (k, N_{min}), where k indicates that the user want that there are at least another $k-1$ peers in cloaked road segments to achieve k-anonymous, L_{min} indicates the minimum acceptable length summation of cloaked road segments, and N_{min} indicates the minimum acceptable number of cloaked road segments. L_{min} and N_{min} are particularly useful in dense area where even a large k can not achieve the user's privacy requirements. For L_{min}, if the length summation of road segments is larger, it will be harder to re-identify the mobile user is on which point of road segments. For N_{min}, if there are more road segments in a cloaked road segments, it will be harder to re-identify the mobile user is on which road segments. Besides, there is a cache in each mobile user's device to decrease the probability of re-identity for the continuous query by reducing the number of queries. At the same time, it can reduce the costs of cloak for the *Spatial-Temporal Connective Location Anonymizer* and query for LBS servers. And we use LRU (Least Recently Used) algorithm for cache replacement.

The *Spatial-Connective Location Anonymizer* receives continuous location updates from mobile users and it blurs the location of any query requesting user q to a cloaked road segments CRS, instead of a spatial region to achieve k-anonymity like [14] or [12], to match user's profile (k, L_{min}) or (k, N_{min}) and forwards the privacy protected query to the location-based service providers. For example, given a spatial network $G(V, E)$ like Figure 3.2, where $V = \{n_1, n_2, n_3, n_4, n_5\}$ and $E = \{(n_1, n_2), (n_2, n_3), (n_2, n_4), (n_2, n_5)\}$. There are three different mobile users which are u_1, u_2 and u_3, and their locations are $u_1(x_1, y_1)$, $u_2(x_2, y_2)$ and $u_3(x_3, y_3)$ respectively. Assumed $k = 3$, after bluring them into cloaked road segments, they will have the same location tuple which is $\{(n_1, n_2), (n_2, n_3), (n_2, n_4)\}$

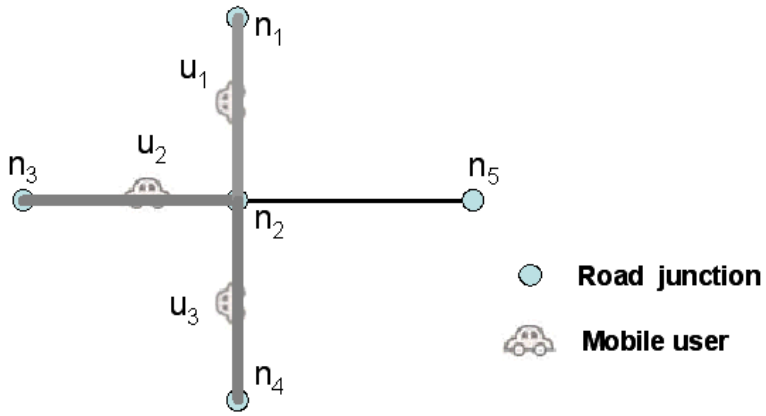


Figure 3.2: An example of cloaked road segments.

and form 3-anonymity. Note that any user identity related information in the query is also removed by the *Spatial-Temporal Connective Location Anonymizer* during the cloaking process.

The *Privacy-Protected Query Processor* is included by location-based service provider to handle privacy-protected query. Instead of returning an exact answer, the privacy-protected query processor returns the candidate answer list for query requesters through the *Spatial-Temporal Connective Location Anonymizer*. After receiving the result set, mobile users can distill the exact answers by their locations in linear time. The privacy profile of a user determines the computational complexity of their spatial queries. Strict privacy requirements (i.e., large k, L_{min} and N_{min} values) increase the complexity of processing the query. So, mobile users have the ability to adjust the personal trade-off by their privacy profiles. In order to improve efficiency, only cloaked spatial queries have to be processed by the *Privacy-Protected Query Processor*, non-cloaked queries can be processed with existing spatial query algorithms.

We assume a digitization process that translates an input spatial network into a modeling graph and storages the modeling graph in the above three entities of our architecture. The nodes of the modeling graph are generated by the road junctions, the start/end points of road segments, and other subsidiary points such as speed

limit change points. In addition, as describe in [18], we assume the network storage scheme will propose the three entities of our architecture different operations:

- For mobile user:
  - $NDist\,(p_1, p_2)$: returns the network distance of two input points p_1,p_2 in the network by applying a algorithm such as Dijkstra's algorithm [3] to compute the shortest path from p_1 to p_2.
- For the *Spatial-Temporal Connective Location Anonymizer*:
  - $find\_segment\,(p_x)$: returns the road segment which user x locates on.
  - $Number\_of\_user\,(segment_i)$: returns the number of user on $segment_i$.
- For the *Privacy-Protected Query Processor*:
  - $find\_objects\,(segment_i)$: returns the data objects which covered by $segment_i$.
  - $NDist\,(p_1, p_2)$: returns the network distance of two input points p_1,p_2 in the network by applying a algorithm such as Dijkstra's algorithm [3] to compute the shortest path from p_1 to p_2.

Table 3.1 collects the symbolic notation used throughout this paper.

Description	**Symbol**
Minimum number of road segments in cloaked road segments	N_{min}
Minimum length summation of road segments in cloaked road segments	L_{min}
Level h	L_h
Total number of road junctions in a spatial network	$Total_{junction}$
Total length summation of all road segments in a spatial network	$Total_{length}$
Total number of road segments in a spatial network	$Total_{seg}$
Number of blocks in level h	$N\_block_{L_h}$
Number of mobile users in block i of L_h	$N\_user_{B_{h,i}}$
Length summation of road segments in block i of L_h	$Length_{B_{h,i}}$
Number of segments in block i of L_h	$N\_seg_{B_{h,i}}$
Network Distance from a to b	$NDist(a, b)$
Hierarchical index structure of spatial network for L_{min}	$Index_{length}$
Hierarchical index structure of spatial network for N_{min}	$Index_{Num\_segment}$

Table 3.1: Description of symbols.

Chapter 4

The Spatial-Temporal Connective Location Anonymizer

The function of the location anonymizer is that it can blur each mobile user's exact location into a cloaked road segments. We think a better location anonymizer needs to satisfy the following three requirements:

- **Accuracy.** If the cloaked road segments satisfies and as close as possible to a user's privacy profile (i.e. k, L_{min}, N_{min}), the location anonymizer will obtain better accuracy.
- **Efficiency.** If the time complexity of cloaked algorithm and the cost of maintenance are as low as possible, the location anonymizer will obtain better efficiency.
- **Flexibility.** The cloaked road segments can satisfy different user's requirements. And each user can change their privacy profile at any time.

In order to satisfy the above requirements as much as possible, we propose our *Spatial-Temporal Connective Location Anonymizer*. Section 4-1 shows the data structure of *Spatial-Temporal Connective Location Anonymizer*. Section 4-2 shows the maintenance of the data structure. Section 4-3 shows the cloaking algorithm.

4.1 Data structure

The data structure of the *Spatial-Temporal Connective Location Anonymizer* is shown in Figure 4.1. We propose this hierarchical structure to bottom up compose those road segments into different L_h level and form different granularity until the root which has only one block covers the whole spatial network. When $h = 0$, each road segments will be each one block. So, there are $Total_{seg}$ blocks in L_0 level where $Total_{seg}$ is the total number of road segments in a spatial network. When $h > 0$, each block of L_{h-1} will merge with its' neighbor to obtain larger blocks of L_h and get lower granularity. In the example of Figure 4.1, edge (n_{10}, n_4) and edge (n_{10}, n_9) of L_0 merge together and form block $B_{1,1}$ of L_1. Next, Block $B_{1,1}$ and $B_{1,2}$ of L_1 merge together and form $B_{2,3}$ of L_2. $B_{2,1}$ and $B_{2,3}$ of L_2 merge together and form $B_{3,1}$ of L_3. $B_{3,1}$ and $B_{3,2}$ of L_3 merge together and form $B_{4,1}$ of L_4. Each block is represented as $\left(B_{h,i}, N\_user_{B_{h,i}}, Length_{B_{h,i}}, N\_seg_{B_{h,i}}\right)$ where $B_{h,i}$ is the block identifier of L_h, $N\_user_{B_{h,i}}$ is the number of users in $B_{h,i}$, $Length_{B_{h,i}}$ is the length summation in $B_{h,i}$ and $N\_seg_{B_{h,i}}$ is the number of segments in $B_{h,i}$. By this hierarchical structure, we can quickly find the cloaked road segments to fit user's profile (k, L_{min}) or (k, N_{min}).

According to user's profile is (k, L_{min}) or (k, N_{min}), we can build two different index tree to match each of them better. If user's profile is (k, L_{min}), we will build the index tree named $Index_{length}$ which can let the length summation of the cloaked road segments is as close as possible to L_{min}. If the summation of road length in each block of the same level is as the same as possible, the quality of spatial network partition is better. Given a spatial network, there are $Total_{junction}$ road junctions, and $Total_{length}$ length summation of all road segments. We define

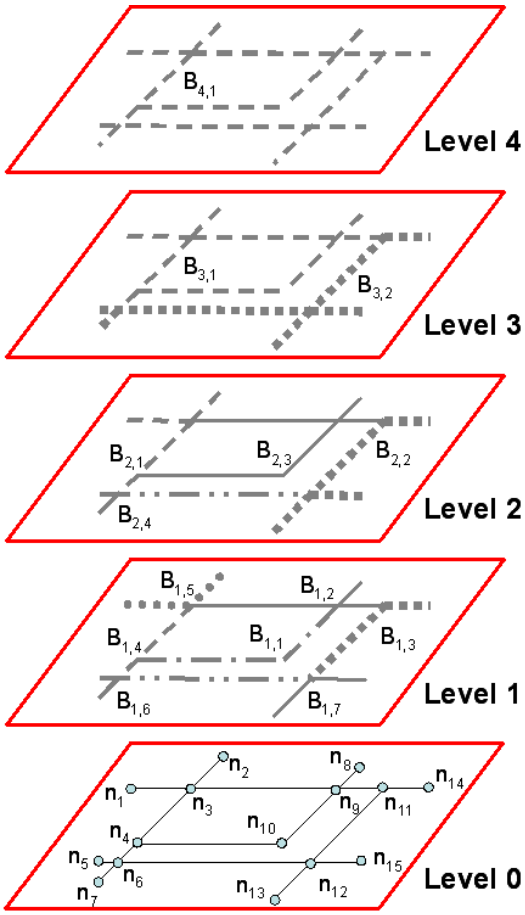


Figure 4.1: An example of hierarchical structure of the Spatial-Temporal Connective Location Anonymizer.

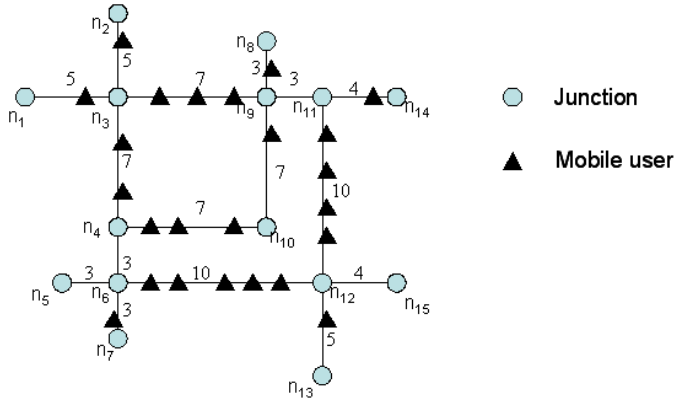


Figure 4.2: An example of spatial network with mobile users and length of each segment.

the spatial network partition quality as

$$\left\{ \begin{array}{l} \sum_{i=1}^{N\_block_{L_h}} \left| Length_{B_{h,i}} - \frac{Total_{length}}{Total_{junction}} \right|, \text{ if } h = 1 \\ \sum_{i=1}^{N\_block_{L_h}} \left| Length_{B_{h,i}} - \frac{Total_{length}}{\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil} \right|, \text{ if } h > 1 \end{array} \right\}$$

. For level 1, we merge the road segments to their common junction, so the maximum number of blocks in level 1 is $Total_{junction}$ and the minimum average length summation of road segments in each block of level 1 is $\frac{Total_{length}}{Total_{junction}}$. For higher level, if we want to merge the blocks from L_{h-1} to L_h, the maximum number of blocks in level h is $\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil$ and the minimum average length summation of road segments in each block of level h is $\frac{Total_{length}}{\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil}$. The length summation of road segments in block i of level h is $Length_{B_{h,i}}$. We calculate the variance between $Length_{B_{h,i}}$ and $\frac{Total_{length}}{Total_{junction}}$, when $h = 1$, or $Length_{B_{h,i}}$ and $\frac{Total_{length}}{\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil}$, when $h > 1$, to recognize if the length summation of road segments are averagely distributed in those blocks. The less the variance is, the higher the quality of partition is.

The $Build\_Index_{length}$ algorithm is shown in Algorithm 1. Assume the spatial network is shown in Figure 4.2. At first, we calculate the total length summation

Algorithm 1 $Build\_Index_{length}$ Algorithm

Input: A modeling spatial network graph, $G(V, E)$

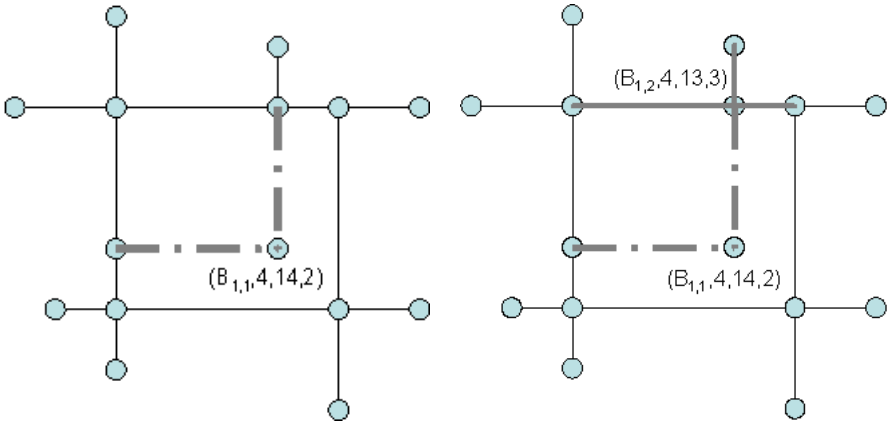
Output: A hierarchical index structure, $Index_{length}$, for L_{min}

$Total_{length}$= the total length summation of road segments in $G(V, E)$
$Total_{junction}$= the total number of road junctions in $G(V, E)$
$k = 0$
while Any segment is not distributed **do**
  Finds junction, n_i,whose length summation of adjacent segments is closest to $\frac{Total_{length}}{Total_{junction}}$
  n_i's adjacent segments become $B_{1,i}$ and remove those segments from n_i's adjacent node,n_j
  Re-sum up the adjacent length summation and number of adjacent segments of n_j
  $k++$
end while
$N\_block_{L_1} = k$
$h = 2$
while $N\_block_{L_{h-1}} > 1$ **do**
  $k = 0$
  while Any $B_{h-1,i}$ have not been merged **do**
    if $B_{h-1,i}$ connects to $B_{h-1,j}$ and they have not merge with other block **then**
      if $Length_{B_{h-1,i}} + Length_{B_{h-1,j}}$ is closet to $\frac{Total_{length}}{\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil}$ **then**
        Merge $B_{h-1,i}$ with $B_{h-1,j}$ to $B_{h,t}$
        $k++$
      end if
    end if
  end while
  $N\_block_{L_h} = k$
  $h++$
end while

of all road segments and the number of junction (Line 1 and 2 in Algorithm 1). In this example, the length summation of all road segments is 86 and the number of junctions is 7. Next, we start to merge the blocks from level 0 to 1 according to the above spatial network partition quality:

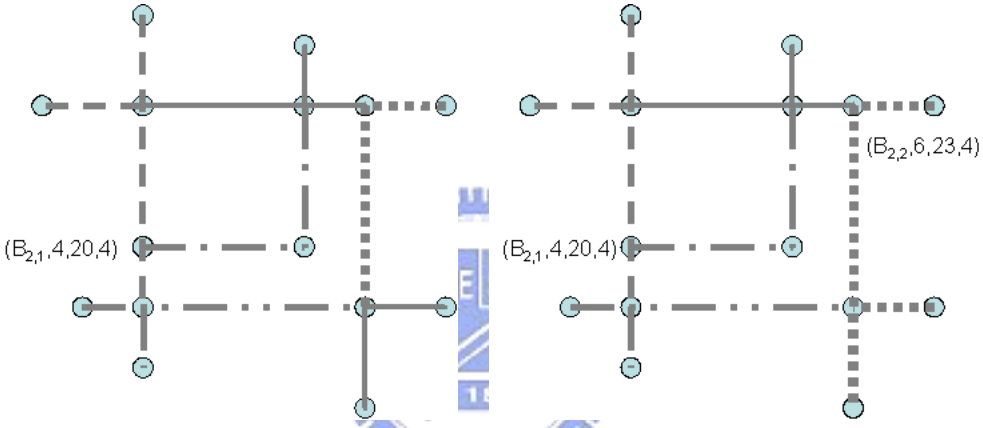
$$\left\{ \begin{array}{l} \sum_{i=1}^{N\_block_{L_h}} \left| Length_{B_{h,i}} - \frac{Total_{length}}{Total_{junction}} \right|, \text{ if } h = 1 \\ \sum_{i=1}^{N\_block_{L_h}} \left| Length_{B_{h,i}} - \frac{Total_{length}}{\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil} \right|, \text{ if } h > 1 \end{array} \right\}$$

. We greedily choose the junction whose length summation of road segments is the closet to $\frac{Total_{length}}{Total_{junction}}$. Once the road segments have already been distributed to a road junction, it can not be distributed to another junction again. So, the length summation of road segments in each junction will change after each choosing procedure (Line 4 to 9 in Algorithm 1). In our example, we will choose the junction whose length summation of road segments is the closet to $\frac{86}{7} = 12.3$. We first choose the junction n_{10} whose length summation is 14, shown in Figure 4.3(a), the adjacent edges of n_{10} merge together and form $B_{1,1}$. Then it causes the road segments (n_{10}, n_4) and (n_{10}, n_9) need be removed from the adjacent list of n_4, n_9 respectively, so the remain length summation of junction n_4 and n_9 will become 10 and 13 respectively. So, shown in Figure 4.3(b), the next choosen junction is n_9 and form $B_{1,2}$. Repeat the procedure until all of the road segments have been distributed to their junctions. Level 1 of Figure 4.1 shows the result of L_1. Next, we choose the pair of blocks in L_{h-1} whose length summation of road segments is the closet to $\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil$ and merge together to obtain a larger block of L_h. Repeats the merge procedure until there exists only one block that covers the whole spatial network (Line 12 to 24 in Algorithm 1). In our example, $N\_block_{L_1} = 7$, so the maximum average length summation of road segments in each block of L_2 is $\frac{86}{\left\lceil \frac{7}{2} \right\rceil} = 21.5$. We find if B_4 merge with B_5, the new block's length summation



(a) First step of merge L_0 (b) Second step of merge L_0

Figure 4.3: Merge step of L_0



(a) First step of merge L_1 (b) Second step of merge L_1

Figure 4.4: Merge step of L_1

of road segments will be 20 which is closet to 21.5. So, we merge B_4 with B_5 and obtain B_8 of L_2, shown in Figure 4.4(a). Next, shown in Figure 4.4(b), we merge B_3 with B_7 and obtain B_9 of L_2. Repeat the procedure until there are no blocks of L_2 can be merge. Level 2 of Figure 4.1 shows the result of L_2. Repeat the same procedure of each level, we can obtain the result of L_3 and L_4 and we obtain the index structure, $Index_{length}$ shown in Figure 4.1.

If user's profile is (k, N_{min}), we will build the index tree named $Index_{Num\_segment}$ which can let the number of road segments in the cloaked road segments is as close as possible to N_{min}. The concept of building $Index_{Num\_segment}$ is almost the

same with $Index_{length}$, and the difference of them is that the partition quality of $Index_{Num\_segment}$ is according to

$$\left\{ \begin{array}{l} \sum_{i=1}^{N\_block_{L_h}} \left| N\_seg_{B_{h,i}} - \frac{Total_{seg}}{Total_{junction}} \right|, \text{ if } h = 1 \\ \sum_{i=1}^{N\_block_{L_h}} \left| N\_seg_{B_{h,i}} - \frac{Total_{seg}}{\left\lceil \frac{N\_block_{L_{h-1}}}{2} \right\rceil} \right|, \text{ if } h > 1 \end{array} \right\}$$

.

4.2 Maintenance

The mobile user will periodically update his location to the *Spatial-Connective Location Anonymizer* in this form (u_{id}, x, y) where u_{id} is used to identify this mobile user, x and y are the spatial coordinate of this mobile user's new location. Once the *Spatial-Temporal Connective Location Anonymizer* receives this update information, it will apply the $find\_segments(p_x)$ operation to find this mobile user locates on which road segment, $segment_{new}$. Then the *Spatial-Temporal Connective Location Anonymizer* will check the original road segment, $segment_{old}$, with $segment_{new}$. If $segment_{old} = segment_{new}$, there is no additional processing. If $segment_{old} \neq segment_{new}$, the number of users on both $segment_{old}$ and $segment_{new}$ need to update. Then propagate the $Number\_of\_user\,(segment_{old})$ and $Number\_of\_user\,(segment_{new})$ in the block counters $N\_user_{B_{h,i}}$ of their higher layer. If a new mobile user registers to our architecture, the number of users on the segment which this user locates on needs to increase by one and update the block counters $N\_user_{B_{h,i}}$ of its' higher layer.

4.3 The Spatial-Temporal Connective Cloaking Algorithm

Algorithm 2 *Spatial-Temporal Connective Cloaking* Algorithm

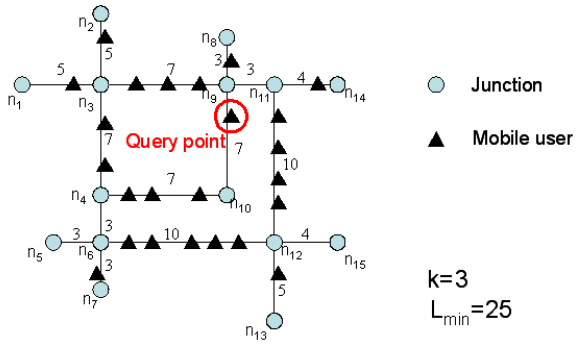
Input: User's profile (k, L_{min}), location (x, y) and hierarchical index structure $Index_{length}$

Output: Cloaked road segments

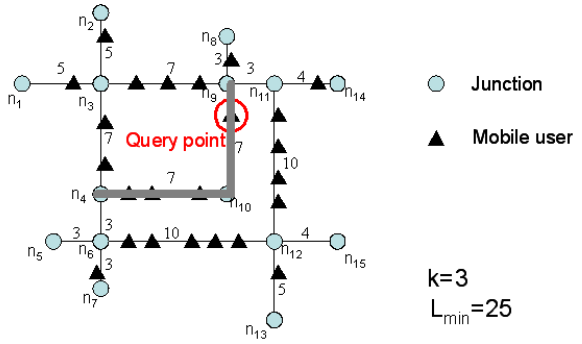
1: Find the road segment, (n_i, n_j), which (x,y) locates on
2: segment (n_i, n_j) belongs to $B_{0,t}$
3: h=0
4: **while** $N\_block_{L_h} \geq 1$ **do**
5: **if** $N\_user_{B_{h,t}} \geq k$ && $Length_{B_{h,t}} \geq L_{min}$ **then**
6: return the road segments covered by $B_{h,t}$
7: **end if**
8: find block $B_{h+1,s}$ which is the parent of block $B_{h,t}$
9: h++
10: t=s
11: **end while**

Algorithm 2 bottom-up blurs a mobile user's location into cloaked road segments. Whenever user wants ask some service to LBS, the *Spatial-Temporal Connective Location Anonymizer* will apply this cloaking algorithm to obtain cloaked road segments and protect user's location privacy. For simplicity, we assume that the query user's profile is in (k, L_{min}) format, so the input of the algorithm is the user's privacy profile k, L_{min}, his location (x, y) and the hierarchical index structure $Index_{length}$. If the query user's profile is in (k, N_{min}) format, it is only to replace $Index_{length}$ with $Index_{Num\_segment}$, L_{min} with N_{min} and $Length_{B_{h,t}}$ with $N\_seg_{B_{h,t}}$ in Algorithm 2. Assume there is a spatial network with length of road segments and mobile users and one of the users query for a service, shown in Figure 4.5(a). This user's privacy profile is $k = 3$ and $L_{min} = 25$. First, the *Spatial-Connective Location Anonymizer* will find out the query user locates on which block of L_0 (Line 1 to 2 in Algorithm 2). Next, it will bottom-up check the blocks, $B_{h,t}$, if the number of users in $B_{h,t}$ is larger than k and the length summation in $B_{h,t}$ is

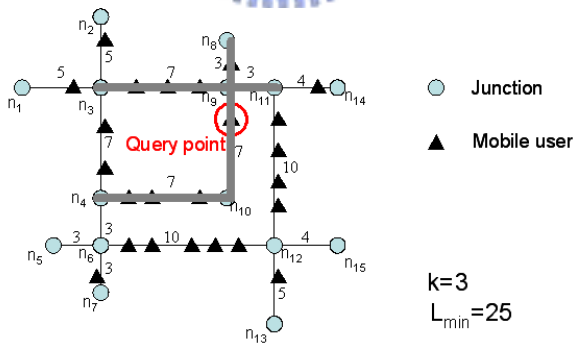
larger than L_{min} (Line 4 to 11 in Algorithm 2). If yes, return the road segments covered by $B_{h,t}$. In our example, shown in Figure 4.5(a), the query user location on segment (n_9, n_{10}) whose length is 7 and there is only one user on it. Because 7 is less than L_{min} and 1 is less than k, it needs to merge with other road segments. Then the *Spatial-Temproal Connective Location Anonymizer* will bottom-up find $B_{1,1}$, shown in Figure 4.5(b), whose length summation is 14 and there are 4 users on $B_{1,1}$. Although 4 is larger than k, but 14 is less than L_{min}. So, it needs to merge with more road segments. The *Spatial-Temproal Connective Location Anonymizer* bottom-up finds $B_{2,3}$, shown in Figure 4.5(c), whose length summation is 27 which is larger than L_{min} and there are 8 users which is larger than k on $B_{2,3}$. So, the *Spatial-Connective Location Anonymizer* will return the cloaked road segments which is $\{(n_9, n_3), (n_9, n_8), (n_9, n_{11}), (n_{10}, n_4), (n_{10}, n_9)\}$.



(a) Cloaks user's location in L_0



(b) Cloaks user's location in L_1



(c) Cloaks user's location in L_2

Figure 4.5: Bottom-up cloak user's location.

Chapter 5

Privacy-Protected Query Processing

In our architecture, shown in Figure 3.1, the *Privacy-Protected Query Processor* can deal with the private queries over public data on spatial networks. It uses the cloaked road segments rather than the exact location to obtain the candidate answers. The traditional query processing algorithm on spatial networks, such as INE or RNE [18], can only support the public queries over public data because the location of query point must be available. So, we illustrate the algorithm for the *Privacy-Protected Query Processor* to solve the private queries. Without loss of generality, we focus on k nearest neighbor query.

When the *Private-Protected Query Processor* receives the privacy-protected k nearest neighbor query from the location anonymizer, it will apply the *Privacy-Protected k Nearest Neighbor Query* algorithm to deal with the cloaked road segments. Given a spatial network with target objects and cloaked road segments CRS, the *Privacy-Protected k Nearest Neighbor Query* algorithm will first find out all of the objects within CRS. Those objects must be the partial candidate answers (Line 1 in Algorithm 3). Next, the algorithm will find out all of the intersection nodes between CRS and the spatial network (Line 4 in Algorithm 3).

Algorithm 3 *Privacy-Protected k Nearest Neighbor Query* Algorithm

Input: A spatial network with interest objects and cloaked road segments *CRS*

Output: Candidate answers

1: $Candidate_{ans}$ = all of objects in *CRS*
2: $NDist_{max} = \infty$
3: Declare a priority queue Q
4: Find the intersection nodes between *CRS* and underlying network as T=$\{t_1, t_2, ..., t_n\}$
5: **for** each t_i **do**
6: $\{O_1, O_2, ..., O_k\}$ = the k nearest object of t_i sorted in ascending order of their network distance (The initial value of $O_1, O_2, ..., O_k$ are all $\emptyset$)
7: $n = t_i$
8: **while** $NDist(t_i, n) < NDist_{max}$ **do**
9: **for** each non-visited adjacent node n_i of n **do**
10: $Cover_{object} = find\_objects(n, n_i)$
11: update $\{O_1, O_2, ..., O_k\}$ from $Cover_{object}$
12: $NDist_{max} = NDist(t_i, O_k)$ (if $O_k = \emptyset, NDist_{max} = \infty$)
13: en-queue $(n_i, NDist(t_i, n_i))$ to Q
14: **end for**
15: de-queue the next node n in Q
16: **end while**
17: $Candidate_{ans} = Candidate_{ans} \cup \{O_1, O_2, ..., O_k\}$
18: **end for**

For each intersection node t_i, it searches its' k nearest neighbor objects and stores in the set of $\{O_1, O_2, ..., O_k\}$. The algorithm will incremental expand to find the k nearest neighbor objects from t_i (Line 8 to 16 in Algorithm 3). $\{O_1, O_2, ..., O_k\}$ are also the partial candidate answers. Finally, union the objects in CRS and $\{O_1, O_2, ..., O_k\}$ of each intersection node t_i, the result set is the candidate answers of k nearest neighbor objects of CRS (Line 17 in Algorithm 3).

For example, Figure 5.1 shows a spatial network with four target objects, O_1, O_2, O_3 and O_4, and the cloaked road segments $\{(n_{10}, n_4), (n_{10}, n_9)\}$. Assume $k = 2$, and we want to find the 2 nearest neighbor objects of the cloaked road segments $\{(n_{10}, n_4), (n_{10}, n_9)\}$. First, the objects within CRS that is O_2 must be the partial candidate answers. $Candidate_{ans}$ is $\{O_2\}$. Next, the intersection nodes between CRS and this spatial network are n_4 and n_9. Then the *privacy-protected k nearest neighbor query* algorithm will search the 2 nearest neighbor objects of n_4 and n_9. For n_4, it fist visits its' adjacent node n_3 and find out O_1 on segments (n_4, n_3). Because the $NDist(n_4, O_1) = 3$, $NDist_{\max}$ becomes 3. Then n_3 pushes to the priority queue Q. $Q =< (n_3, 7) >$. Next, n_4 visits n_6 and n_{10} and pushes them to Q. $Q =< (n_6, 3), (n_3, 7), (n_{10}, 7) >$. Next de-queue n_6 from Q, n_6 will visit its' adjacent node n_5, n_7 and n_{12}. It finds out O_3 is on the segments (n_6, n_{12}). So, $\{O_1, O_3\}$ is the candidate list of 2 nearest neighbor objects of n_4. Because O_3 is the second nearest neighbor object of n_4, $NDist_{\max}$ becomes 11. And $Q =< (n_5, 6), (n_7, 6), (n_3, 7), (n_{10}, 7), (n_{12}, 13) >$. Because n_5 and n_7 cannot be expanded, it de-queues n_5, n_7 and n_3 from Q and n_3 will visit its' adjacent node n_1, n_2 and n_9. $Q =< (n_{10}, 7), (n_1, 12), (n_2, 12), (n_{12}, 13), (n_9, 14) >$. Next de-queue n_{10} from Q, n_{10} will visit its' adjacent node n_9. It finds out O_2 is on the segments (n_{10}, n_9). So, $\{O_1, O_2\}$ is the candidate list of 2 nearest neighbor objects of n_4 and $NDist_{\max}$ becomes 9. Because the network distance from n_4 to the other nodes in Q are larger than $NDist_{\max}$, the privacy-protected k nearest neighbor query

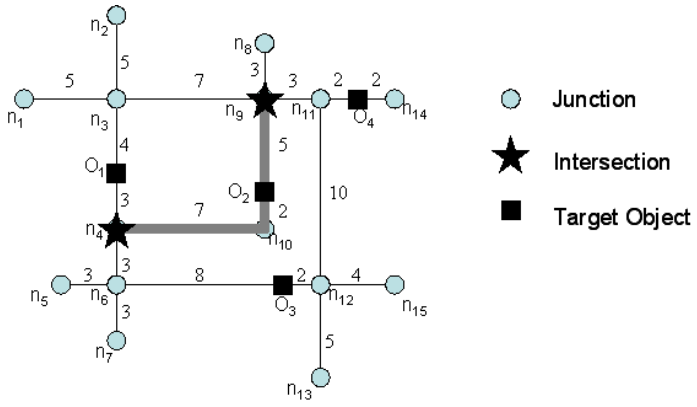


Figure 5.1: Privacy-protected query for the k nearest neighbor objects of cloaked road segments. ($k = 2$ in this example)

algorithm will stop to search the other nearest neighbor of n_4. The same with n_4, the candidate list of 2 nearest neighbor objects of n_9 is $\{O_2, O_4\}$. Finally, union $\{O_1\}$, $\{O_1, O_2\}$ and $\{O_2, O_4\}$ will obtain the candidate list of 2 nearest neighbor object of $\{(n_{10}, n_4), (n_{10}, n_9)\}$ that is $\{O_1, O_2, O_4\}$.

Chapter 6

Experiments

In this section, we will evaluate the performance of our system by evaluating its' three components that are mobile user, *Spatial-Temporal Connective Location Anonymizer* and *Privacy-Protected Query Processor*. In all of our experiments, we use the *Network-based Generator of Moving Objects* [2] to generate moving objects. We use its' attached Oldenburg's road map data file, shown in Figure 6.1, as the input to the generator. The generator will output a set of moving objects that move on the road network of the given map. We set there are 5000 mobile users on the spatial network and they will update their location per time stamp. Next, Target objects are randomly distributed on the spatial network. We assume that each edge exists at most only one target object.

6.1 Mobile user

In this subsection, we evaluate the overall performance of our *Spatial-Temporal Connective Cloaking* algorithm with respect to the cache hit rate for mobile users to prove the contribution of this work by evaluating three different influences that are effect of user's condition, effect of user's profile and effect of query. We randomly choose one mobile user as the query point who issues an k-nearest-neighbor

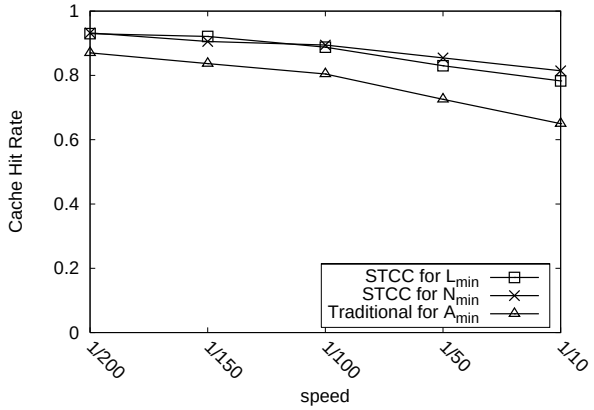
Figure 6.1: Map of Oldenburg.

continuous query which persists 30 time stamps and there are 3000 target objects in this spatial network.

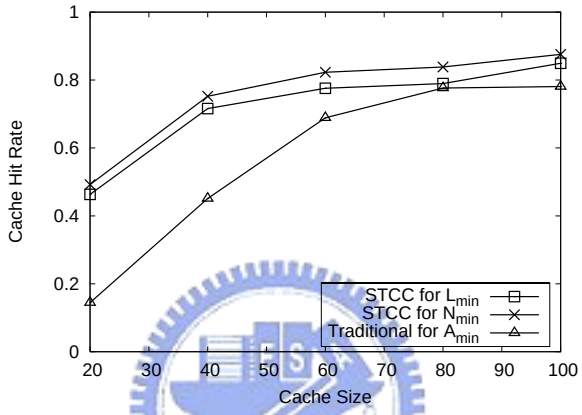
6.1.1 Effect of user's condition

Figure 6.2(a) compares the effect of speed to our cloaked road segments with the effect to traditional cloaked region by increasing speed from $\frac{1}{200}$ to $\frac{1}{10}$ (the denominator is the speed parameter in the generator). We assume user sets his privacy profile as 30-anonymity and L_{min} and N_{min} are both of 0, he issues one-nearest-neighbor query and his cache size is 100. By increasing the speed, the cache hit rate decreases because user will easily leave out of our cloaked road segments or the traditional cloaked region. Even though, the cache hit rate of our cloaked road segments is higher than traditional cloaked region especially for higher speed.

Figure 6.2(b) compares the effect of cache size to our cloaked road segments with the effect to traditional cloaked region by increasing the size from 20 to 100. We assume user sets his privacy profile as 30-anonymity and L_{min} and N_{min} are



(a) Effect of speed



(b) Effect of cache size

Figure 6.2: Effect of user's condition.

both of 0, he issues one-nearest-neighbor query and his speed is $\frac{1}{50}$ (middle in the generator). By increasing the cache size, the cache hit rate increases because the cache on mobile device can cache more record in it. Even though, the cache hit rate of our cloaked road segments is higher than traditional cloaked region especially for less cache size.

6.1.2 Effect of user's profile

Figure 6.3 compares the effect of k-anonymity in user's privacy profile to our cloaked road segments with the effect to traditional cloaked region by increasing k from 10 to 100 while L_{min} and N_{min} are both of 0. We assume user issues one-nearest-

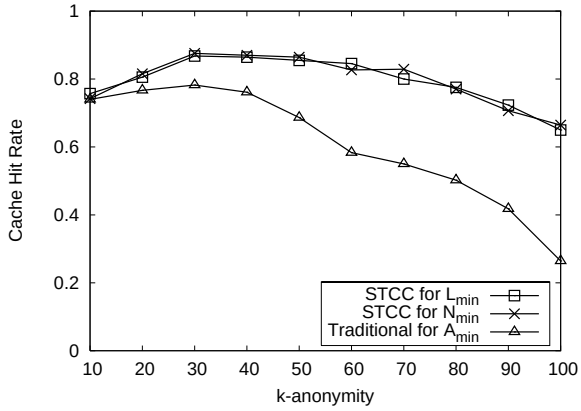
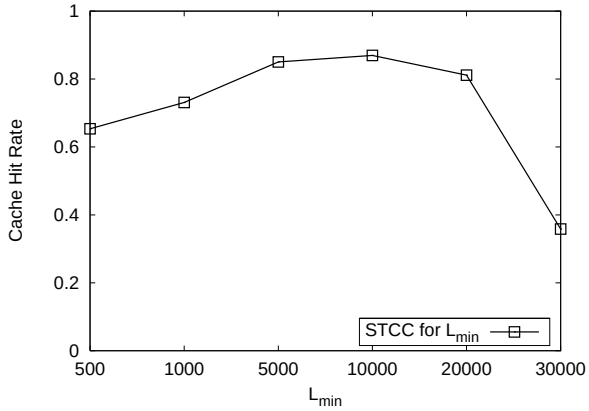


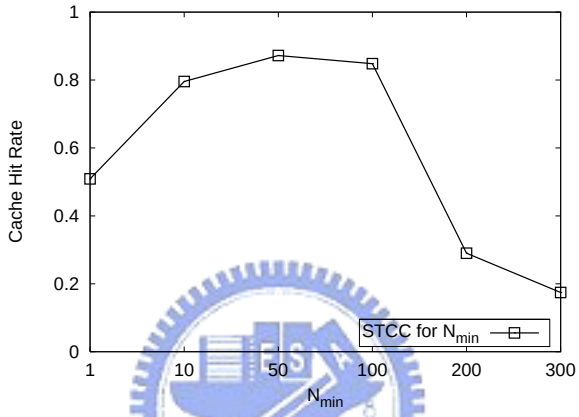
Figure 6.3: Effect of k.

neighbor query, his speed is $\frac{1}{50}$ and his cache size is 100. By increasing the k value of k-anonymity, the cache hit rate of our cloaked road segments and traditional cloaked region increase from $k = 10$ to $k = 30$, but decrease from $k = 40$ to $k = 100$. Because the cache of user's device can cache larger length summation or number of segments for cloaked road segments or larger region for traditional cloaked region and its' candidate answer size is not over the cache size until $k = 30$. When $k > 30$, its' candidate answer size will begin to over the cache size and cause cache miss. Even though, the cache hit rate of our cloaked road segments is higher than traditional cloaked region especially for larger k.

Figure 6.4 gives the effect of L_{min} and N_{min} in user's privacy profile to our cloaked road segments by increasing L_{min} from 500 to 30000 and N_{min} from 1 to 300 while the k value of k-anonymity is 1. We assume user issues one-nearest-neighbor query, his speed is $\frac{1}{50}$ and his cache size is 100. By increasing the L_{min} or N_{min}, the cache hit rate increases from $L_{min} = 500$ to $L_{min} = 10000$ or from $N_{min} = 1$ to $N_{min} = 50$, but decreases from $L_{min} = 10000$ or $N_{min} = 50$. Because the cache of user's device can cache larger length summation or number of segments for cloaked road segments and its' candidate answer size is not over the cache size until $L_{min} = 10000$ or $N_{min} = 50$. When $L_{min} > 10000$ or $N_{min} > 50$, its' candidate answer size will begin to over the cache size and cause cache miss. Even though,



(a) Effect of L_{min}



(b) Effect of N_{min}

Figure 6.4: Effect of L_{min} and N_{min}.

the cache hit rate of our cloaked road segments is higher than traditional cloaked region especially for larger L_{min} or N_{min}.

6.1.3 Effect of query

Figure 6.5 compares the effect of k value of k-nearest-neighbor query to our cloaked road segments with the effect to traditional cloaked region by increasing the k value of k-nearest-neighbor from 1 to 20. We assume user sets his privacy profile as 30-anonymity and L_{min} and N_{min} are both of 0, his speed is $\frac{1}{50}$ and his cache size is 100. By increasing the k value of k-nearest-neighbor, the cache hit rate decreases because the privacy-protected query processor will return more candidate answers and those result the candidate answer size easily over the cache size and cause

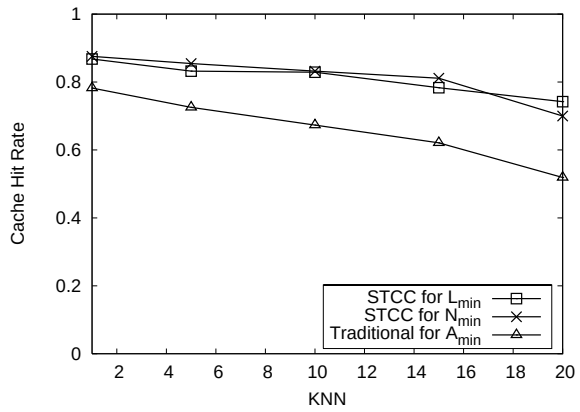


Figure 6.5: Effect of query.

cache miss. Even though, the cache hit rate of our cloaked road segments is higher than traditional cloaked region especially for larger k value of k-nearest-neighbor.

Through all of the above experiments in 6.1 subsection, we prove that our *Spatial-Temporal Connective Cloaking* algorithm gives higher cache hit rate that helps to reduce the re-identified probability of query point and the costs of cloaking and query for continuous query on spatial networks.

6.2 Spatial-Temporal Connective Location Anonymizer

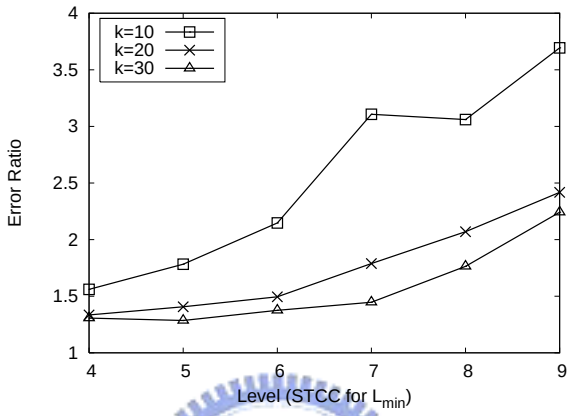
In this subsection, we evaluate the efficiency of the *Spatial-Temporal Connective Location Anonymizer* with respect to accuracy because user wants to have a cloaked road segments that can best match his privacy profile. However, the *Spatial-Temporal Connective Location Anonymizer* may not give an exact match because of the resolution of the hierarchical structure. We randomly choose one mobile user as the query point and assume he issues an one-nearest-neighbor continuous query which persists 30 time stamps, his speed is $\frac{1}{50}$(middle in this generator) and the cache size of mobile device is 100 candidate answers. Besides, there are 3000 target objects in this spatial network.

6.2.1 k Precision

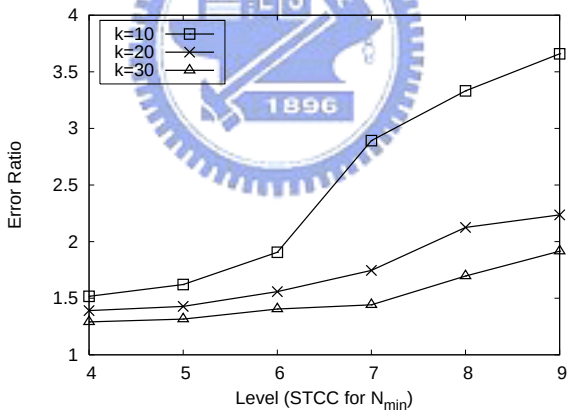
The accuracy is measured as $\frac{k'}{k}$, where k' is the number of users in cloaked road segments while k is the exact user requirement in his privacy profile. We run the experiment for different privacy requirement, k = 10, 20 and 30, while L_{min} and N_{min} are 0. Figure 6.6(a) and Figure 6.6(b) give the effect of the hierarchical level of two different index tree $Index_{length}$ and $Index_{Num\_segment}$ respectively on k precision of the cloaked road segments. Lower hierarchical levels give very accurate answer whose error ratio is very close to one (optimal case) whether for relaxed users or not. Higher hierarchical levels give less accurate answer especially for relaxed users. Because higher hierarchical level provides lower resolution that means there are fewer blocks and more mobile users in each block. This results the number of users in the block of higher hierarchical level more easily over k too much and cause higher error ratio. Besides, for relaxed users, their k values are small, so even a little difference will cause large error ratio.

6.2.2 L_{min} and N_{min} Precision

The accuracy is measured as $\frac{L'_{min}}{L_{min}}$ and $\frac{N'_{min}}{N_{min}}$, where L'_{min} and N'_{min} are respectively the length summation and number of segments in cloaked road segments while L_{min} and N_{min} are the exact user requirements in his privacy profile. For L_{min}, we run the experiment for different privacy requirement, L_{min} = 1000, 2000 and 3000. For N_{min}, we run the experiment for different privacy requirement, N_{min} = 10, 15 and 20. Both of the two experiments, we set k to 0. Figure 6.7(a) gives the effect of the hierarchical level of the index tree, $Index_{length}$, on L_{min} precision of the cloaked road segments. Lower hierarchical levels give very accurate answer whose error ratio is very close to one (optimal case) whether for relaxed users or not. Higher hierarchical levels give less accurate answer especially for relaxed users. Because higher hierarchical level provides lower resolution that means there are fewer blocks

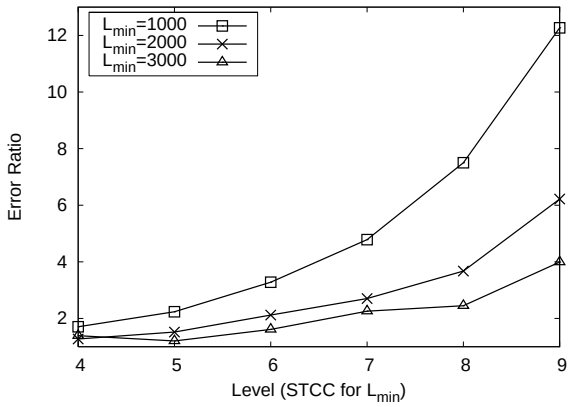


(a) k precision for L_{min}

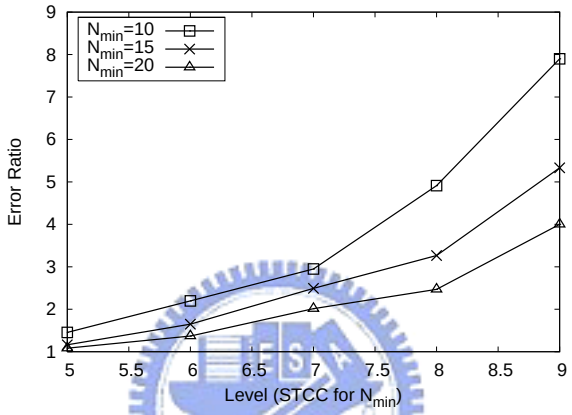


(b) k precision for N_{min}

Figure 6.6: k precision.



(a) L_{min} precision



(b) N_{min} precision

Figure 6.7: L_{min} and N_{min} precision.

and larger length summation in each block. This results the length summation in the block of higher hierarchical level more easily over L_{min} too much and cause higher error ratio. Besides, for relaxed users, their L_{min} values are small, so even a little difference will cause large error ratio. The same as L_{min}, Figure 6.7(b) gives the effect of the hierarchical level of the index tree, $Index_{Num\_segment}$, on N_{min} precision of the cloaked road segments.

6.3 Privacy-Protected Query Processor

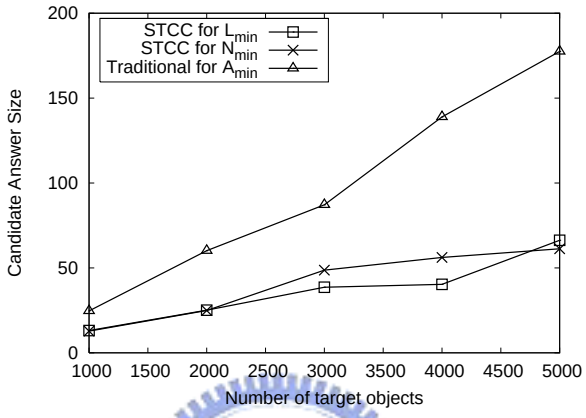
In this subsection, we evaluate the scalability and efficiency of the *Privacy-Protected Query Processor* with respect to the returned candidate answer size because we

want to prove if our cloaked road segments can obtain less candidate answer size than traditional cloaked region and raise the cache hit rate. We randomly choose one mobile user as the query point and assume he issues an k-nearest-neighbor continuous query which persists 30 time stamps, his speed is $\frac{1}{50}$(middle in this generator) and the cache size of mobile device is 100 candidate answers.

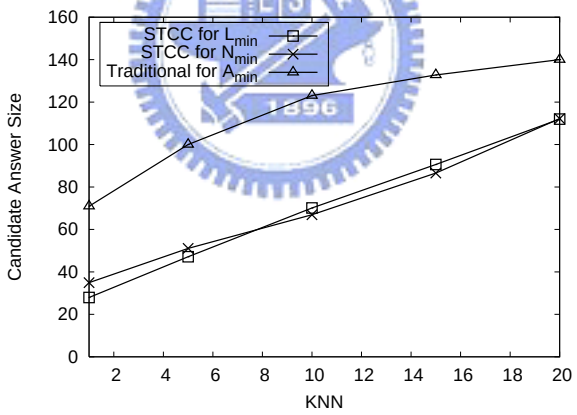
6.3.1 Scalability

Figure 6.8(a) compares the scalability of our cloaked road segments with traditional cloaked region by increasing the number of target objects from 1000 to 5000. We assume user sets his privacy profile as 30-anonymity and L_{min} and N_{min} are both of 0 and he issues one-nearest neighbor query. By increasing the number of target objects, the candidate answer size of our cloaked road segments and traditional cloaked region increase because the more target objects in the spatial network, the more target objects will be covered by cloaked road segments and traditional cloaked region. Even though, the candidate answer size of our cloaked road segments is less than traditional cloaked region especially for large target objects.

Figure 6.8(b) compares the scalability of our cloaked road segments with traditional cloaked region by increasing the k value of k-nearest-neighbor query from 1 to 20. We assume user sets his privacy profile as 30-anonymity and L_{min} and N_{min} are both of 0 and the target objects in the spatial network are 3000. By increasing the k value of k-nearest-neighbor query, the candidate answer size of our cloaked road segments and traditional cloaked region increase because it needs to find more neighbor target objects. Even though, the candidate answer size of our cloaked road segments is less than traditional cloaked region especially for large target objects.



(a) Number of target objects Scalability



(b) KNN Scalability

Figure 6.8: Scalability.

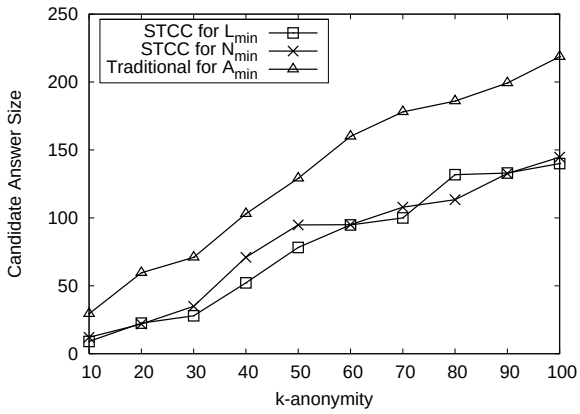
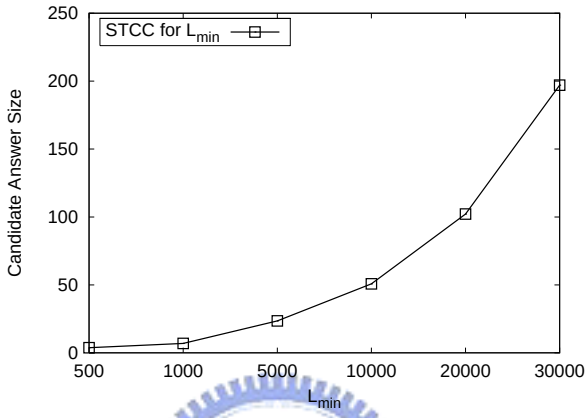


Figure 6.9: Effect of k.

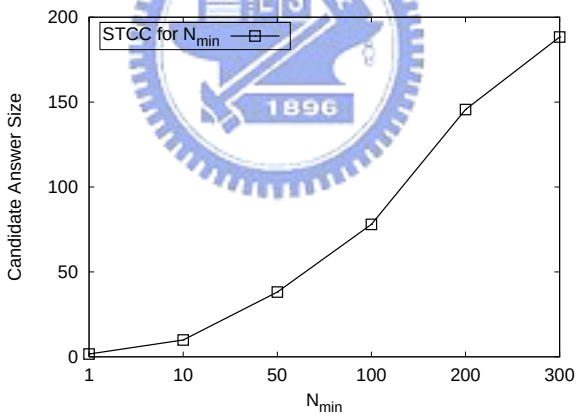
6.3.2 Effect of privacy profile

Figure 6.9 compares the effect of k-anonymity in user's privacy profile to our cloaked road segments with the effect to traditional cloaked region by increasing k from 10 to 100 while L_{min} and N_{min} are both of 0. We assume there are 3000 target objects in the spatial network and user issues one-nearest-neighbor query. By increasing the k value of k-anonymity, the candidate answer size of our cloaked road segments and traditional cloaked region increase because it needs larger number of segments or region for larger k and this will result that there are more target objects will be covered by cloaked road segments and traditional cloaked region. Even though, the candidate answer size of our cloaked road segments is less than traditional cloaked region especially for large target objects.

Figure 6.10 gives the effect of L_{min} and N_{min} in user's privacy profile to our cloaked road segments by increasing L_{min} from 500 to 30000 and N_{min} from 1 to 300 while the k value of k-anonymity is 1. We assume there are 3000 target objects in the spatial network and user issues one-nearest-neighbor query. By increasing the L_{min} or N_{min}, the candidate answer size increases too because it needs larger number of segments for larger L_{min} or N_{min} and this will result that there are more target objects will be covered by cloaked road segments.



(a) Effect of L_{min}



(b) Effect of N_{min}

Figure 6.10: Effect of L_{min} and N_{min}.

Chapter 7

Conclusion

This paper introduces a system with three components that are mobile users, *Spatial-Temporal Connective Location Anonymizer* and *Privacy-Protected Query Processor* to efficiently process continuous query without compromising privacy and large overheads. First, we implement cache mechanism on mobile user's device to reduce the number of queries. Mobile users can set their privacy profile (k, L_{min}) or (k, N_{min}). In order to efficiently utilize the cache, we design *Spatial-Temporal Connective Cloaking* algorithm in the S*patial-Temporal Connective Location Anonymizer*. The *Spatial-Temporal Connective Location Anonymizer* can produce two different hierarchical index trees that are $Index_{length}$ and $Index_{Num\_segment}$ to best match user's profile (k, L_{min}) and (k, N_{min}) respectively. The hierarchical index structure can blur query point into different granularity cloaked road segments. Next, the *Privacy-Protected Query Processor* can process the privacy-protected query according to the cloaked road segments and return the candidate answer list to query point through the *Spatial-Temporal Connective Location Anonymizer*. Finally, we experimentally evaluate our system by evaluating the performance of mobile user, *Spatial-Temporal Connective Location Anonymizer* and *Privacy-Protected Query Processor* and prove that our system can efficiently process continuous query without compromising privacy and large overheads.

Bibliography

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

[2] T. Brinkhoff. A Framework for Generating Network-Based Moving Objects. *GeoInformatica*, 6(2):153–180, 2002.

[3] E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[4] B. Gedik and L. Liu. "Location Privacy in Mobile Systems: A Personalized Anonymization Model". In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS), Columbus, OH, USA*, 2005.

[5] M. Gruteser and D. Grunwald. "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking". In *Proceedings of the First ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, USA*, 2003.

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

[7] G. R. Hjaltason and H. Samet. Distance Browsing in Spatial Databases. *ACM Trans. Database Syst.*, 24(2):265–318, 1999.

[8] H. Hu, J. Xu, and D. L. Lee. "A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects". In *Proceedings of the 2005 ACM International Conference on Management of Data (SIGMOD), Baltimore, Maryland, USA*, 2005.

[9] N. Jefferies, C. J. Mitchell, and M. Walker. "A Proposed Architecture for Trusted Third Party Services". In *Proceedings of the International Conference on Cryptography: Policy and Algorithms, Brisbane, Queensland, Australia*, 1995.

[10] C. S. Jensen, D. Lin, B. C. Ooi, and R. Zhang. "Effective Density Queries on Continuously Moving Objects". In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE), Atlanta, GA, USA*, 2006.

[11] M. R. Kolahdouzan and C. Shahabi. "Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases". In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB), Toronto, Canada*, 2004.

[12] W.-S. Ku, R. Zimmermann, W.-C. Peng, and S. Shroff. "Privacy Protected Query Processing on Spatial Networks". In *Proceedings of the third IEEE International Workshop on Privacy Data Management (PDM), (In conjunction with the 2007 IEEE International Conference on Data Engineering), Istanbul, Turkey*, 2007.

[13] M. F. Mokbel. "Towards Privacy-Aware Location-Based Database Servers". In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE) Workshops, Atlanta, Georgia, USA*, 2006.

[14] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. "The New Casper: Query Processing for Location Services without Compromising Privacy". In *Proceedings*

of the 32nd International Conference on Very Large Data Bases (VLDB), Seoul, Korea, 2006.

[15] M. F. Mokbel, X. Xiong, and W. G. Aref. "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases". In *Proceedings of the 2004 ACM International Conference on Management of Data (SIGMOD), Paris, France*, 2004.

[16] K. Mouratidis, M. Hadjieleftheriou, and D. Papadias. "Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring". In *Proceedings of the 2005 ACM International Conference on Management of Data (SIGMOD), Baltimore, Maryland, USA*, 2005.

[17] K. Mouratidis, M. L. Yiu, D. Papadias, and N. Mamoulis. "Continuous Nearest Neighbor Monitoring in Road Networks". In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB), Seoul, Korea*, 2006.

[18] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. "Query Processing in Spatial Network Databases". In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB), Berlin, Germany*, 2003.

[19] A. Pfitzmann and M. Kohntopp. "Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology". In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA*, 2000.

[20] N. Roussopoulos, S. Kelley, and F. Vincent. "Nearest Neighbor Queries". In *Proceedings of the 1995 ACM International Conference on Management of Data (SIGMOD), San Jose, California*, 1995.

[21] B. N. Schilit, J. I. Hong, and M. Gruteser. Wireless Location Privacy Protection. *IEEE Computer*, 36(12):135–137, 2003.

[22] L. Sweene. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[23] T. Xia and D. Zhang. "Continuous Reverse Nearest Neighbor Monitoring". In *Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, USA*, 2006.

[24] M. L. Yiu, N. Mamoulis, and D. Papadias. Aggregate Nearest Neighbor Queries in Road Networks. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):820–833, 2005.

[25] X. Yu, K. Q. Pu, and N. Koudas. "Monitoring K-Nearest Neighbor Queries Over Moving Objects". In *Proceedings of the 21nd International Conference on Data Engineering (ICDE), Tokyo, Japan*, 2005.