2012 International Symposium on Pervasive Systems, Algorithms and Networks

Load Balancing with Cell Breathing in EPON-WiMAX Integrated Network

Hung-Chi Lai
Dept. of Computer Science and Information Engineering
National Chung-Cheng University
Chiayi, Taiwan, R.O.C
lhc98m@cs.ccu.edu.tw

Hung-Yi Teng
Dept. of Computer Science and Information Engineering
National Chung-Cheng University
Chiayi, Taiwan, R.O.C
thy95p@cs.ccu.edu.tw

Ren-Hung Hwang
Dept. of Computer Science and Information Engineering
National Chung-Cheng University
Chiayi, Taiwan, R.O.C
rhhwang@cs.ccu.edu.tw

Abstract —Multimedia applications over the Internet, such as IPTV and video on demand (VOD), have become fast growing applications in recent years. Such applications have stringent QoS constraints in terms of bandwidth, delay and packet loss. As a consequence, broadband access networks play an important role for multimedia applications. There are two emerging technologies offering both high bandwidth and QoS support, Ethernet Passive Optical Network (EPON) and Worldwide Interoperability for Microwave Access (WiMAX). By integrating these two technologies, EPON-WiMAX integrated network can: (1) provide broadband access, (2) support mobile users, and (3) decrease network planning cost and operating cost. Thus, EPON-WiMAX integrated network is an ideal choice for multimedia applications with ubiquitous access. In the EPON-WiMAX integrated network, ONU-BSs send the data received from the OLT to their subscribe stations (SSs) with QoS guarantee. However, some ONU-BSs could become hot-spots due to dynamics of mobility and bandwidth requirement of SSs. When an ONU-BS becomes overloaded, the QoS requirements would not be guaranteed and therefore the overall performance would be significantly decreased. In this study, a load balancing mechanism based on cell breathing was proposed for the hybrid EPON-WiMAX network. In the proposed approach, the load balancing problem was formulated into a linear programming problem. Our goal is to find the best power adjustment that maximizes system throughput. We present a Heaviest Load First Algorithm (HLFA) to obtain the optimal solution. We demonstrate the performance of our approach via extensive simulations. The simulation results show that the HLFA can provide the best solution to achieve load balancing and enhance the system throughput even though the system has multiple overloaded ONU-BSs.

Keywords—EPON-WiMAX integrated network, load balancing, cell breathing

I. INTRODUCTION

In recent years, a variety of multimedia applications have posted a high bandwidth requirement, such as high-definition television (HDTV) and video-on-demand (VOD) services, which indicated that the broadband access technology has become more and more important. In wired networks, Ethernet Passive Optical Network (EPON) [1] is a point-to-multipoint (P2MP) optical fiber access network which supports up to 10Gbps bandwidth. Since it can provide high bandwidth and is compatible with legacy Ethernet, it is considered as one of the solutions for the next-generation wired broadband access technologies. In wireless networks, Worldwide Interoperability for Microwave Access (WiMAX) [2][3] is a new generation of broadband wireless access technology which supports long-distance, high-bandwidth, QoS-guaranteed wireless communications. Therefore, it has been identified as one of the last mile solutions. Although EPON and WiMAX technologies are promising, it should be noted that deploying EPON or WiMAX networks still has some limitations. For example, deploying Fiber To The Home (FTTH) is expensive for Internet Service Providers (ISPs). On the other hand, the data-transfer rate of a mobile WiMAX station in current real world implementation may only be maxing out around 45 Mbps/channel which is much less than that of wired networks.

Therefore, Shen et al. [4] proposed the integration of EPON and WiMAX networks to make up for each other's deficiencies. Advantages of the integrated EPON-WiMAX network include providing broadband Internet access, supporting mobile users, and reducing the network design and maintenance costs. In general, the architecture of the EPON-WiMAX network is a tree topology where an EPON Optical Line Terminal (OLT) is connected to several EPON Optical Network Units (ONUs) which consequently connects to a WiMAX Base Station (BS). Based on how EPON ONU and WiMAX BS are constituted, the architecture was classified into four categories, namely independent, hybrid, unified, and microwave-over-fiber. Among these four architectures, the hybrid architecture was the most promising with advantages of more flexible development, less deployment costs, and less technology restrictions. In the hybrid architecture, EPON ONU and WiMAX BS were integrated into one device, referred to as ONU-BS. An ONU-BS consists of three components: EPON ONU, WiMAX BS and a central control unit. The central control unit is responsible for conversion between EPON and WiMAX networks, such as frame format conversion and QoS mapping. A MultiPoint Control Protocol (MPCP) is used for communication between OLT and ONU-BS which is defined in IEEE 802.3ah [5].

In the EPON-WiMAX integrated network, ONU-BSs send the data received from the OLT and provide QoS guarantee to their subscribe stations (SSs). Since SSs have mobility capacity and different bandwidth demand, some ONU-BSs may become overloaded in a certain time period. Under such conditions, the QoS requirements of SSs of the overloaded ONU-BS will not be guaranteed and therefore the overall performance would be significantly decreased. To balance the traffic load at the heavily loaded ONU-BSs, a well-known approach is the use of cell breathing [6]. The basic idea of cell breathing is to dynamically adjust the coverage of a cell to reduce or increase loads based on networking information. However, how to find the appropriate power assignment to the ONU-BSs to achieve load balancing and performance improvement is crucial. Relatively less attention has been focused on load balancing for the EPON-WiMAX integrated network. *Wong et al.* [7] proposed a load balancing mechanism based on cell breathing technique. In the mechanism, an iterative algorithm is presented that equalizes the expected transmission time according to the reported queue sizes and adjusted transmission power level.

1087-4089/12 $26.00 © 2012 IEEE
DOI 10.1109/I-SPAN.2012.32

In this paper, a load balancing mechanism based on cell breathing is proposed for the hybrid EPON-WiMAX network. In the proposed approach, the load balancing problem is formulated into a linear programming problem. Our goal is to find the best power adjustment that maximizes system throughput. Then, we present a Heaviest Load First Algorithm (HLFA) to obtain the optimal solution. Using the HLFA algorithm, the transmission power of ONU-BS can be adaptively adjusted based on the amount of transmitted data and SSs' channel condition. In the HLFA algorithm, power level of the heaviest-loaded ONU-BS and/or its neighbors is adjusted to achieve both load balancing and throughput improvement. More importantly, the solution obtained by the HLFA algorithm will not waste any transmission power to make the system energy-efficient. We demonstrate the performance of our approach via extensive simulations. The simulation results show that the HLFA provides the best solution to achieve load balancing and improves the system throughput even through the system has multiple overloaded ONU-BSs.

The rest of the paper is organized as follows. Section II gives an overview of our system model. In Section III, we propose a Heaviest Load First Algorithm (HLFA) to balance the load and improve the system throughput. Simulation results are presented in Section IV. Finally, conclusion and future work are given in Section V.

II. System Model

A. Network Environment

As shown in Figure 1, there are three main components in an integrated EPON-WiMAX network. OLT, a component of EPON, receives packets from the wired network and then transfers them to ONU-BSs. The ONU-BS is an integrated component of the ONU of EPON and the BS of WiMAX. It receives packets from the EPON wired network and then transmits them to SSs through the WiMAX network. SS is a component of WiMAX which receives packets only from the serving ONU-BS. In this study, we focus on the load balancing problem at ONU-BSs. Thus, in our integrated EPON-WiMAX network environment, the OLT plays a role of decision maker which is responsible for selecting the best combination of transmitted power level based on the amount of SSs' data and SSs' SINR value. We make the following assumptions to clarify the load balancing problem.

1. There are no coverage holes among ONU-BSs when all ONU-BSs adopt the lowest transmission power level.
2. Each ONU-BS is able to access full context of all SSs which are under its coverage, including distance and signal strength. It passes the information of SS to OLT using MPCP.
3. The OLT know the neighboring relationship among ONU-BSs.
4. The frequency of each ONU-BS is orthogonal to that of its neighboring ONU-BSs so that the channel interference is negligible.

In addition, an SS could be a mobile station. Therefore, an SS could handoff from one ONU-BS to another. There are three handoff mechanisms supported in IEEE 802.16e, namely, Hard Handover (HHO), Macro-Diversity Handover (MDHO), Fast Base Station Switching (FBSS). For delay-sensitive applications, MDHO and FBSS are better mechanisms [8]. In this study, we assumed the FBSS mechanism is adopted because of its flexibility. Coordinated by OLT, ONU-BSs assist an SS to associate the designated ONU-BS for better network throughput.

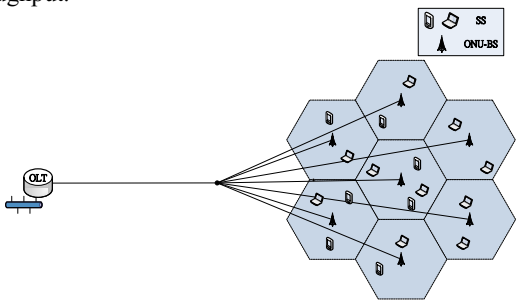


Figure 1. EPON-WiMAX network topology

B. Problem formulation

Next, we formulate the load balancing problem into a linear programming problem. Let the number of ONU-BSs and the number of SSs be M and N, respectively. Each ONU-BS has K levels of transmission power, namely, from P_1 to P_K where P_1 is the lowest power level and P_K is the highest power level. Let D_i be the amount of data sent from the OLT to SS_i in a predefined period. Since data rate is different when D_i is transmitted by different power level or ONU-BS, we use $Ts(m, k, n)$ and $M(m, k, n)$ to represent the number of time slots and the number of bits can be transmitted in a slot when the ONU-BS m adopts the power level k to transmit the data to the SS_n. Then we consider how to select the best power level for each ONU-BS and which ONU-BS should transmit the data for the SS_n. Let P_{mk} and A_{mn} be

$$P_{mk} = \begin{cases} 1, & \text{if } ONU\text{-}BS_m \text{ uses power level } k \text{ transmission} \\ 0, & otherwise \end{cases} \quad \textbf{(1)}$$

$$A_{mn} = \begin{cases} 1, & \text{if } ONU\text{-}BS_m \text{ is designated to send data to } SS_n \\ 0, & otherwise \end{cases} \quad \textbf{(2)}$$

Since each ONU-BS can transmit data by only using a single power level and a SS can associate only one ONU-BS at the same time, the power level constraint of an ONU-BS m and the association constraint of a SS n are as follows.

$$\sum_{k=1}^{K} P_{mk} = 1 \quad \forall m = 1,..,M \quad \textbf{(3)}$$

$$\sum_{m=1}^{M} A_{mn} \leq 1 \quad \forall n = 1,..,N \quad \textbf{(4)}$$

Furthermore, the number of time slots in a WiMAX OFDMA frame is limited. Therefore, we have the following resource constraint:

$$\sum_{k=1}^{K} \sum_{n=1}^{N} Ts(m,k,n) P_{mk} A_{mn} \leq Ts\_MAX, \quad \textbf{(5)}$$

where Ts_MAX denotes the resource budget of ONU-BSs. Our objective is to maximize the system throughput while the overall power level could be minimized. By changing the power level of ONU-BSs and the associated ONU-BS of SSs, we have

$$\text{Maximum } \left(\sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{n=1}^{N} Ts(m,k,n) M(m,k,n) P_{mk} A_{mn} - \sum_{m=1}^{M} \sum_{k=1}^{K} \frac{k}{MK+1} \times P_{mk}\right) \quad \textbf{(6)}$$

There may be several adjustments of power levels that provide the same throughput; therefore, we use $\sum_{m=1}^{M} \sum_{k=1}^{K} \frac{k}{MK+1} \times P_{mk}$ to find the minimal power level adjustment. Table 1 shows the notation used in this study.

Table 1. List of Notations

Notations	Meaning
M	The total number of ONU-BS in the system
K	The total number of power levels used in the ONU-BSs
N	The total number of SS in the system
D_i	The amount of data sent from the OLT to SS_i in a predefined period
Ts(m,k,n)	The number of time slots when the ONU-BS *m* adopts the power level *k* to transmit the data to the SS_n
M(m,k,n)	The number of bits can be transmitted in a slot when the ONU-BS *m* adopts the power level *k* to transmit the data to the SS_n
P_{mk}	The indicator that whether ONU-BS_m use the power level k to transmit
A_{mn}	The indicator that whether SS_n is associated with ONU-BS_m

Since Equations (5) and (6) are nonlinear, we defined a 0/1 variable Z_{mkn} and replace Equation (7) by Equation (8) and (9) as follows.

$$Z_{mkn} = P_{mk}A_{mn} = \begin{cases} 1, & \text{if ONU-BS}_m \text{ transmits data to SS}_n \text{ at power level k} \\ 0, & \text{otherwise} \end{cases} \quad \textbf{(7)}$$

$$-Z_{mkn} + P_{mk} + A_{mn} \leq 1 \quad \textbf{(8)}$$

$$2Z_{mkn} - P_{mk} - A_{mn} \leq 0 \quad \textbf{(9)}$$

Finally, we have

$$\text{Maximum } (\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} Ts(m,k,n)M(m,k,n)Z_{mkn} - \sum_{m=1}^{M}\sum_{k=1}^{K}\frac{k}{MK+1} \times P_{mk})$$

$$\sum_{k=1}^{K}\sum_{n=1}^{N} Ts(m,k,n)Z_{mkn} \leq Ts\_MAX \quad \forall m = 1,..,M$$

III. Proposed Algorithm

To solve this linear programming problem, we can use some utilities like CPLEX to obtain the optimal solution. However, the computational complexity of such approaches is not acceptable for delay-sensitive applications. Therefore, we propose a novel algorithm, Heaviest Load First Algorithm (HLFA), to achieve load balancing and throughput improvement. First of all, we define L_m to be the traffic load on an individual ONU-BS *m* and L_m is calculated as follows.

$$L_m = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N} Ts(m,k,n)P_{mk}A_{mn}}{Ts_{MAX}} \quad \textbf{(10)}$$

Note that we have the calculation of the load on each ONU-BS to be based on the amount of transmitted data such that the proposed load balancing mechanism could be more effective.

A. Heaviest load first algorithm (HLFA)

The HLFA algorithm is executed once every scheduled frame on the OLT to adaptively control power level of ONU-BSs. Figure 2 shows the pseudo code of the HLFA algorithm. First, the power level of all ONU-BSs is set to the minimal level in order to avoid waste of transmission power. Then, for each SS, the ONU-BS that has the highest SNR value is selected to be its associated ONU-BS. After that, the load on each ONU-BS can be calculated through CalculateLoad algorithm and the overloaded ONU-BSs are put into the overloaded list. The detailed CalculatedLoad algorithm is described in Section B. If there are overloaded ONU-BSs in the overloaded list, the FindMaxGain algorithm will be triggered to balance the load from the ONU-BS that has the heaviest load first to improve the system throughput. The FindMaxGain algorithm is described further in Section C. The above procedure is repeated until there is no overloaded ONU-BS or the system throughput cannot be improved any more.

Heaviest load first algorithm (HLFA)

```
1  Heaviest_Load_First() {
2      All ONU-BSi.power are initialized to level 1 ;
3    Determine the associated ONU-BS for each SS based SINR;
4    ClaculateLoad(Overloaded_List);
5      While (Overloaded_List is not Empty) {
6       get first ONU-BSx from Overloaded_List;
7       FindMaxGain (ONU-BSx);}
8  }
```

Figure 2. HLFA algorithm

B. CalculateLoad algorithm

The CalculateLoad algorithm is used to calculate the load on each ONU-BS and add the overloaded ONU-BSs into the overloaded list. Figure 3 shows the pseudo code of the CalculateLoad algorithm. Since the association between ONU-BSs and all SSs has been determined, the load and throughput of each ONU-BS can be computed by using Equation (10). At line 4, the variable adjustLevel is initialized to 0 for each ONU-BS. Later on, in FindMaxGain algorithm, this variable will be used to indicate how many power levels need to be increased for an ONU-BS to achieve load balancing. At line 5, if the calculated load of an ONU-BS is larger than 1, it means that the ONU-BS is overloaded; that is, the ONU-BS has buffered more data than it can transmit within a frame time. Overloaded ONU-BSs are put into the overloaded list which is sorted according to the loading of each ONU-BS in the non-increasing order.

CalculateLoad algorithm

```
1  CalculateLoad(Overloaded_list) {
2     for each i from 1 to M do{
3        Calculate ONU-BSi's load and throughput ;
4        ONU-BSi.adjsutLevel set 0 ;
5        if(ONU-BSi.load > 1) {
6           ONU-BSi add to Overloaded_list ;
7           sort Overloaded_list by load in decreasing order ; }
8     }
9    Return Overloaded_list;
10 }
```

Figure 3. CalculateLoad algorithm

C. FindMaxGain algorithm

When the overloaded ONU-BSs are identified through the CalculateLoad algorithm, we use the FindMaxGain algorithm to perform load balancing. Figure 4 shows the pseudo code of the FindMaxGain algorithm. Figure 5 shows its flow chart. In order to reduce the loading of the overloaded ONU-BSs and improve the system throughput, we have two choices: (1) increasing the transmission power of a overloaded ONU-BS to enhance the transmission rate; (2) increasing the transmission

power of the neighboring ONU-BSs of a overloaded ONU-BS such that some of SSs at the overlapping coverage area (cell edge) can be handovered from the overloaded ONU-BS to its neighboring ONU-BS. In the FindMaxGain algorithm, whether to increase the power level of the heaviest-loaded ONU-BS or its neighboring ONU-BS is determined by the throughput improvement calculated by the CalculateThroughputGain algorithm. The detailed CalculateThroughputGain algorithm is described in Section D. In order to find the best throughput improvement, all possible ways of power adjustments and their throughput improvement are assessed. For example, if there are an overloaded ONU-BS and three neighboring ONU-BSs, effect of power adjustment of these four ONU-BSs will be assessed; thus, the CalculateThroughputGain algorithm will be executed four times to obtain the throughput improvement of the power adjustment of each of these four ONU-BSs. When the system throughput can be improved by one of power adjustments, we set the transmission power of the ONU-BS according to the best power adjustment. After that, some of SSs at the cell edge region may be handovered to the lightly-loaded ONU-BS. After the adjustment, we recalculate the load and throughput of the heaviest-loaded ONU-BS. If the heaviest-loaded ONU-BS becomes a light-loaded ONU-BS, the ONU-BS will be removed from the overloaded list. Otherwise, the overloaded list is sorted by the loading in the non-increasing order again, and the heaviest-loaded ONU-BS will be selected for adjustment in the next iteration. The power level of an ONU-BS is increased one level by one level until it reaches the maximal power level. In this case, the ONU-BS will be removed from the overloaded list.

FindMaxGain algorithm

```
1   FindMaxGain (x) {
2       max_throughput_gain = 0 ;
3       for each ONU-BSx's neighbor as well as itself do {
4         // set Target_ONU-BS to one of the ONU-BSx's neighbors or itself
5           Target_ONU-BS = ONU-BSx OR ONU-BSx.nbr[j] ;
6           throughput_gain = 0 ;
7           powerLevel = Target_ONU-BS.power + ONU-BSx.adjustLevel ;
8           if (powerLevel <= MAX_POWER) {
9               throughput_gain=CalculateThroughputGain(ONU-BSx,
10              Target_ONU-BS,powerLevel) ;
11              if (throughput_gain > max_throughput_gain) {
12                  Record.Target_ONU-BS = Target_ONU-BS;
13                  Record.Target_ONU-BS_load =
14                  Target_ONU-BS.tempload;
15                  Record.Target_ONU-BS_throughput =
16                  Target_ONU-BS.tempthroughput ;
17                  Record.ONU-BSx_load = ONU-BSx.tempload;
18                  Record.ONU-BSx_throughput =
19                  ONU-BSx.tempthroughput;
20                  HandOverList = Target_ONU-BS.handOverList;
21                  Record.powerLevel = powerLevel ;
22                   max_throughput_gain = throughput_gain ;
23              }
24          }
25      } // for end
26      if (max_throughput_gain > 0 ) {
27          ONU-BSx.load = Record.ONU-BSx_load; ;
28          ONU-BSx.throughput = Record.ONU-BSx_throughput ;
29          Target_ONU-BS.power = Record.powerLevel ;
30          Target_ONU-BS.load = Record.Target_ONU-BS_load ;
31          Target_ONU-BS.throughput =
32          Record.Target_ONU-BS_throughput ;
33          Handover all SS in HandOverList to Record.Target_ONU-BS ;
34          if (ONU-BSx.load <= 1)
35              remove ONU-BSx from Overloaded_list;
36          else
37              sort Overloaded_list by loading in decreasing order ;
38      } else {
39  // system throughput is not improved, try to increase power one more level
40          if(ONU-BSx.addLevel < MAX_POEWRLEVEL ) {
41              ONU-BSx.adjustLevel++ ;
42          }else {
43              remove ONU-BSx from Overloaded_list;
44          }
45      }
46  }
```

Figure 4. FindMaxGain algorithm

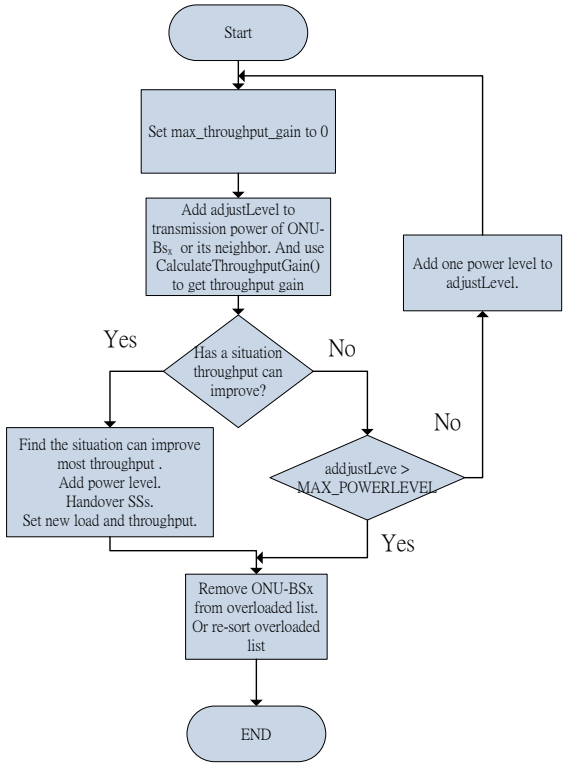


Figure 5. Flow chart of the FindMaxGain algorithm

D. CalculateThroughputGain algorithm

A critical function in the FindMaxGain algorithm is to calculate the throughput improvement for every power adjustment. The calculation is done by the CalculateThroughputGain algorithm. Figures 6 and 7 show the pseudo code and flow chart of the CalculateThroughputGain algorithm, respectively. There are two approaches for performing power adjustments. In the first approach, the heaviest-loaded ONU-BS increases its transmission power to enhance transmission rate for its SSs. For this kind of power adjustments, we simply recalculate the throughput and load of the ONU-BS. The other approach is to increase the transmission power of neighboring ONU-BSs of the heaviest-loaded ONU-BS such that some of SSs at the cell edge can be handovered from the heaviest-loaded ONU-BS to its neighbors. In this kind of power adjustment, the selected neighboring ONU-BS needs to be verified first that it is not overloaded. If the neighboring ONU-BS is not overloaded, the SSs in the overlapping coverage area are put into the handover list and the system throughput and the loading on ONU-BSs involved in the power adjustment are recalculated. In case the neighboring ONU-BS becomes overloaded after the power adjustment, the power adjustment is unacceptable and the algorithm will return a result of 0.

CalculateThroughputGain algorithm

```
1  CalculateThroughputGain(ONU-BSx,Target_ONU-BS, Target_powerLevel) {
2      if(ONU-BSx == Target_ONU-BS) {
3          orignal_throughput = ONU-BSx.throughput ;
4          ONU-BSx.temp_load = CalculateLoad(ONU-BSx,Target_power_level) ;
5          ONU-BSx.temp_throughput =
6          CalculateThroughput(ONU-BSx,Target_power_level) ;
7          throughput_gain = oringnal_throughput – ONU-BSx.temp_throughput ;
8          if(throughput_gain > 0)
9              return throughput_gain ;
10     } else {
11         if(Target_ONU-BS.load < 1) {
12             orignal_throughput = ONU-BSx.throughput+
13             Target_ONU-BS.throughput ;
14             For each SS which associated ONU-BSx
15                 if(the SS is in Target_ONU-BS transmission range with
16                 Target_power_level) {
17                     Add the SS to the Target_ONU-BS.handoverList ;}
18             END For
19             ONU-BSx.temp_load = CalculateLoad(ONU-BSx,ONU-BSx.power,
20             "−",handoverList) ;
21             ONU-BSx.temp_throughput =
22             CalculateThroughput(ONU-BSx,ONU-BSx.power, "−",
23             handoverList) ;
24             Target_ONU-BS.temp_load =
25             CalculateLoad(Target_ONU-BS,Target_power_level,"+",
26             handoverList) ;
27             Target_ONU-BSx.temp_throughput =
28             CalculateThroughput(Target_ONU-BS, Target_power_level,"+",
29             handoverList) ;
30             Throughput_gain = oringnal_throughput –
31             ( ONU-BSx.temp_throughput +
32             Target_ONU-BSx.temp_throughput ) ;
33             if(Throughput_gain > 0 && Target_ONU-BS.temp_load <= 1) {
34                 return throughput_gain ; }
35         }
36     }
37     return 0 ;
38 }
```

Figure 6. CalculateThroughputGain algorithm

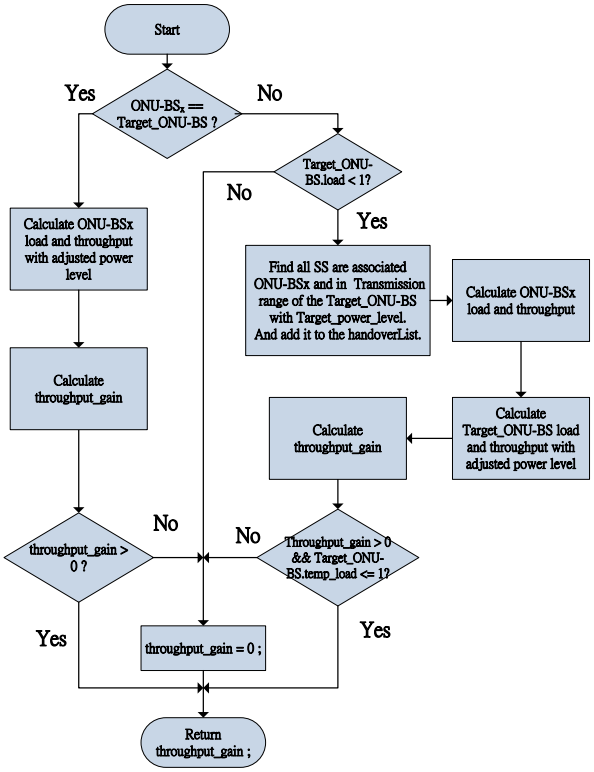


Figure 7. Flow chart of the CalculateThroughputGain algorithm

IV. Numerical Results

A. Simulation Environment

We demonstrate the performance of our approaches via extensive simulations. We use Qualnet to perform the experiments. The network topology of the experiments consists of one OLT and seven ONU-BSs as shown in Fig. 1. The deployment of the ONU-BSs is well planned such that no coverage hole exits when the minimal power level is adopted. Moreover, the frequency of each ONU-BS is orthogonal to that of its neighboring ONU-BSs so that the channel interference is negligible. Each ONU-BS has six power levels and uses the minimal power level to serve its SSs in the initial scenario. We consider a 10 MHz spectrum and set frame duration to 5 ms. Within a frame, the ratio of upper link and down link is set to 1:2. The transmission coverage of the minimal power level is 1000m and increasing a power level will increase 100m transmission range. Table 2 summarizes the WiMAX parameters of the simulations. In addition, the ONU-BS has adaptive modulation and coding schemes (MCSs) capacity and Table 3 shows the available MCSs used in simulations.

Table 2. WiMAX parameters

Parameters	Value
Simulation Time	2000 frames
Physical Layer	OFDMA
Duplex	TDD
Carrier Frequency	3.4 GHz
Channel Bandwidth	10 MHz
FFT Size	1024
Cyclic Prefix Factor	8.0
Frame Duration	5 ms
UL:DL	1:2
BS-to-BS Distance	1.75 kilometers
BS Height	30 meters
SS Height	1.5 meters
Path Loss Model	Two Ray
Shadowing	4.47 dB
Power Level vs. Distance	{1,2,3,4,5,6} vs. {1000,1100,1200,1300,1400,1500}

Table 3. MCSs used in simulations

Modulation	Coding
QPSK	1/2
QPSK	3/4
16-QAM	1/2
16-QAM	3/4
64-QAM	1/2
64-QAM	2/3
64-QAM	3/4

In the simulated EPON-WiMAX network, there are 140 SSs that are strategically distributed in the network as showed in Fig. 8. The ONU-BS 2 serves the largest number of SSs and some SSs are located in the overlapping coverage area (cell edge) of multiple ONU-BSs. We adopt CBR traffics to make input traffics stable and increase the number of CBR traffics to increase the load on the ONU-BSs. We have designed our simulation scenario such that the traffic load of an ONU-BS is continuously increased and finally it also causes traffic overload of multiple ONU-BSs. Specifically, at the beginning of the simulation, all ONU-BS is lightly-loaded. At the time of the 400th frame, the ONU-BS 4 becomes overloaded. Its load is further increased after the time of the 600th frame and the 1200th frame. Furthermore, at the time of the 1600th frame, the ONU-BS 2 also becomes overloaded.

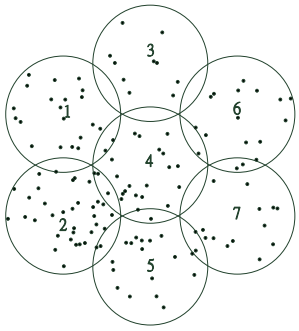


Figure 8. Distribution of 140 SSs

We adopt the system throughput as a performance metric to evaluate transmission efficiency. When the system throughput increases, it indicates that more data can be transmitted in a time unit and therefore the system becomes more efficient. On the other hand, throughput improvement may consume more transmitted power. In this paper, we aim at providing better system throughput with the minimal power consumption. Thus, we also observe the average transmission power per ONU-BS in our simulations. In our simulations, we compared our approach with two approaches: one is that the transmission power of ONU-BSs never changed (NCB); the other is the load balancing algorithm proposed in [9] (NLBA). Since we formulate the load balancing problem into a linear programming problem, we also use CPELX to solve the linear programming problem and compare the result with that of our approach.

B. Simulation result and dissection

First, we investigate the performance of our approach when multiple ONU-BSs become overloaded. Figure 9 shows the system throughput of three approaches in our simulation scenario. In Fig 9, before the 400th frame, all ONU-BSs are lightly-loaded and three approaches provide similar system throughput. After the 400th frame, the ONU-BS 4 becomes overloaded and the performance of three approaches becomes significantly different. The NCB approach provides the worst throughput improvement because it never changes the transmission power of ONU-BSs and the system throughput is limited. In the NLBA approach, the load on an ONU-BS is the summation of the inverse of SS's transmission rate. Therefore, the NLBA approach can provide a better throughput than the NCB approach. However, the system throughput of the HLFA algorithm is higher than other approaches. It is because the HLFA algorithm can take the amount of transmitted data on every frame into account and assess the number of time slots used in transmission. After the 600th frame and the 1200th frame, the ONU-BS 4 has more data to send to its SSs. Again, the HLFA algorithm provides better throughput improvement since it can adaptively adjust power level of ONU-BSs based on the amount of transmitted data on every frame. After the time of the 1600th frame, the ONU-BS 2 also becomes overloaded. Under such condition, the system throughput of the HLFA algorithm is further higher than that of other approaches, which shows that the HLFA algorithm can achieve load balancing as well as throughput improvement effectively.

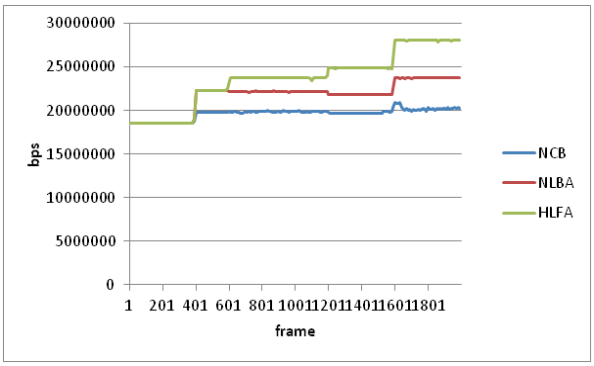


Figure 9. The system throughput of three approaches

Figure 10 shows the average transmission power of three approaches in our simulation scenario. In Fig.10, the transmission power of the NLBA approach is the highest before the 1600th frame even though there are no overloaded ONU-BSs (before the 200th frame). It is because the NLBA approach only considers the transmission rate of SSs and will waste some transmission power for unbalanced SS distribution. However, the HLFA algorithm can properly adjust transmitted power of ONU-BSs based on the actual transmitted data and therefore it can maintain a low transmission power. On the other hand, as shown in Figure 10, to increase system throughput and achieve load balancing, the HLFA algorithm does consume more transmission power than the NCB approach.

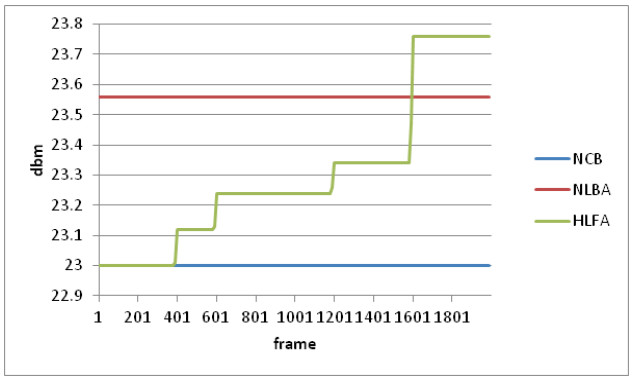


Figure 10. The average transmission power of three approaches

Next, we compare the performance of the HLFA algorithm with the optimal solution obtained by CPLEX. In the following simulations, we design the five scenarios as shown in Table 4. The system throughput and transmission power of yielded/consumed by the HLFA algorithm and the optimal solution is shown in Fig. 11 and Fig. 12, respectively. In Fig.11, the HLFA algorithm provides the same system throughput as the optimal solution (two curves are overlapped in the figure such that they are not differentiable). It is because the HLFA algorithm is able to transmit the data in time for all SSs and does not generate any backlog. In Fig. 12, we can also observe that the power consumption of the HLFA is the same or very close to that of the optimal solution.

Table 4. The simulation scenarios

Scenario	ONU-BS's ID	Load
1	None	<1
2	4	1.30
3	4	1.60
4	4	1.82
5	4, 2	1.95, 1.26

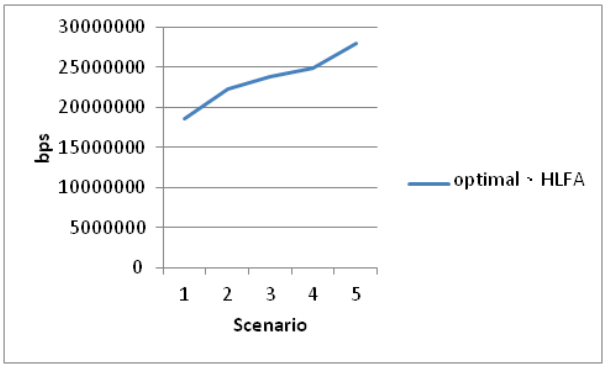


Figure 11. The system throughput of the HLFA algorithm and CLPEX

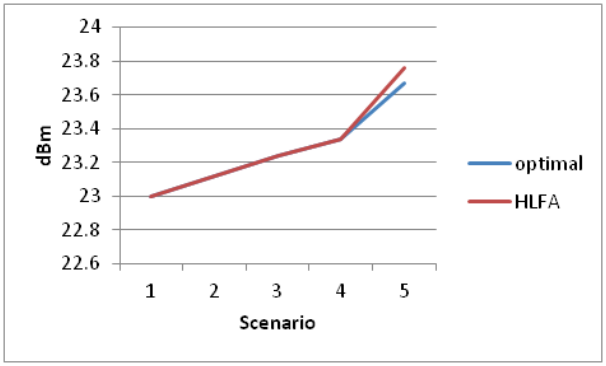


Figure 12. The transmitted power of the HLFA algorithm and CLPEX

V. Conclusion and Future Work

EPON-WiMAX is a promising solution for broadband access network with mobility support. In this study, we focused on the load balancing problem for the ONU-BSs. we proposed a load balancing mechanism based on cell breathing for the hybrid EPON-WiMAX network. In the proposed approach, the load balancing problem was formulated into a linear programming problem. In order to obtain the optimal solution efficiently, we presented a Heaviest Load First Algorithm (HLFA). The HLFA algorithm can adaptively adjust the transmitted power of ONU-BSs based on the amount of transmitted data and SSs' channel condition. More importantly, the solution obtained by the HLFA algorithm also tried to minimize the consumption of transmission power and make the system energy-efficient. We demonstrate the performance of our approach via extensive simulations. The simulation results show that the HLFA provided the best solution to achieve load balancing and enhance the system throughput among three algorithms examined in our simulation scenarios.

In future studies, we are investigating the cooperative transmission among multiple ONU-BSs by using Space Time Coding such that the system throughput can be further improved.

Acknowledgements

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 100－2221－E－194－012－MY3.

References

[1] G. Kramer and G. Pesavento, "Ethernet passive optical network (EPON): building a next-generation optical access network," *IEEE Communications Magazine*, vol. 40, no. 2, pp.66–73, Feb 2002.

[2] IEEE 802.16, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," 2004

[3] IEEE 802.16e, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," Feb. 2005.

[4] G. Shen, R. S. Tucker, and C.-J. Chae, "Fixed Mobile Convergence Architectures for Broadband Access: Integration of EPON and WiMAX," *IEEE Communication Magazine*, vol.45, no.8, pp.44–50, Aug. 2007.

[5] IEEE Std. 802.3ah, "Ethernet in the First Mile," June 2004.

[6] K. A. Ali, H. S. Hassanein and H. T. Mouftah, "A Novel Dynamic Directional Cell Breathing Mechanism with Rate Adaptation for Congestion Control in WCDMA Networks," *IEEE Wireless Communications and Networking Conference*, pp. 2927-2932, Apr.2008.

[7] S.-W. Wong , L. G. Kazovsky , Y. Yan and L. Dittmann, "MPCP assisted power control and performance of cell breathing in integrated EPON-WiMAX network," *IEEE Conference on Global telecommunications,* pp.1-6, 2009.

[8] S. K. Ray, K. Pawlikowski, and H. Sirisena, "Handover in Mobile WiMAX networks: the state of art and research issues," *IEEE Communications Surveys and Tutorials Journal*, vol. 12, no. 3, pp. 376–399, 2010.

[9] L. Yun, J. Hong, L. Xi, and C. Daojin, "A Novel Load Balancing Algorithm in IEEE 802.11 Wireless LANs with Cell Breathing," *Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2009.

2012 International Symposium on Pervasive Systems, Algorithms and Networks

A Novel Time-Obfuscated Algorithm for Trajectory Privacy

Ren-Hung Hwang*, Yu-Ling Hsueh† and Hao-Wei Chung‡
Dept. of Computer Science & Information Engineering
National Chung Cheng University, Taiwan
* *Email: rhhwang@cs.ccu.edu.tw*
† *Email: hsueh@cs.ccu.edu.tw*
‡ *Email: haowei.ck@gmail.com*

***Abstract*—Location-based services (LBS) which bring so much convenience to our daily life have been intensively studied over the years. Generally, an LBS query processing can be categorized into snapshot and continuous queries which search on user location information and reply search results to the users. An LBS has full control to the location information, causing a user privacy concern. If an LBS has a malicious intention to infer the user privacy by tracking the user's routes to their destinations, it incurs a serious problem. In this paper, we propose a comprehensive trajectory privacy technique and combined ambient conditions to cloak location information based on the user privacy profile. We first propose a r-anonymity concept which preprocesses a set of similar trajectories R to blur the actual trajectory of a user. We then combine k-anonymity with s road segments to protect the user privacy. We introduce a novel time-obfuscated technique which breaks the sequence of the query issuing time for a user to confuse the LBS from knowing the user trajectory by sending a query randomly from a set of locations residing at the trajectories R. Despite the randomness incurring from the obfuscation process for providing a strong trajectory privacy protection, the experimental results showed that our trajectory privacy technique maintained the correctness of the query results at a competitive computational cost.**

***Keywords*-Security and Privacy, Mobile Query Processing, Mobile Communication and Computing**

I. Introduction

With the development of GPS equipment, wireless communication technologies and personal mobile devices, location-based services (LBSs) provide the services to users based on their location obtained from their smart phones. For example, OnStar [10] service is a comprehensive LBS system that provides many services to the users such as emergency, navigation, tracking stolen vehicles, and POI searching services. LBSs collects user's location and personal information (e.g., user id) from the query message to retrieve corresponding answers. One major query type to support the location-based services is a continuous query which consists of a set of sequential point queries on the user trajectory. Since users do not know how LBSs manipulate their information, this poses a serious threat to users and raises a privacy concern. For example, a malicious LBS may track continuous queries to retrieve their diary trajectories, from which an LBS may infer their life style, home/work addresses, or even what clinic they had visited in the past.

To protect the trajectory privacy for a user, prior research work proposed some solutions to blur an actual user trajectory. In [13], Xu *et al.* cloaked the footprints on a user trajectory with $(k-1)$ different footprints on the historical trajectories to satisfy the k-anonymity paradigm. In [1], when performing a continuous query, to protect a user's identification from being disclosed by adversaries, Chow *et al.* utilized a k-sharing range issued from every query location to contain k identical users to avoid a malicious retrieval for the actual user trajectory. Unfortunately, existing techniques for protecting trajectory privacy have some weaknesses. Author Xu and Cai [13] adopted historical trajectories to satisfy the trajectory-based k-anonymity paradigm. However, when there is an empty set of users on an anonymous trajectory, a malicious LBS can easily speculate the queries sent by the user and track their trajectories to identify the user. Another issue is that current techniques do not consider the sequence of query issuing time for a continuous query, where each point query issued from a location on the user trajectory is received by an LBS in sequence beginning with the user's starting point and ending with the user's destination. The query location for each point query is transformed into a cloaked region to protect the identity. However, a malicious LBS can still easily track the queries and eventually retrieve a possible user trajectory. For example in Figure 1, for both cases, $R1$ contains the POIs of a park and a shop. $R2$ contains a company and a mall. Each bold arrow line represents the most possible trajectory of a user. An LBS can calculate the probability of all possible paths from the POIs in $R1$ to those in $R2$ based on a user's velocity (i.e., direction and speed) and speculate a possible user path. In this example, the probability of four paths are 0.45, 0.2, 0.25, and 0.1, a malicious LBS may easily infer that bold arrow line is the most possible path of a user.

In this paper, we explore four privacy factors to solve the above-mentioned problems. First, for r-anonymity, we propose a preprocessing technique. When a user starts to plan a travelling route (i.e., a user trajectory), the trusted server obtains $(r-1)$ history trajectories similar to the user trajectory from databases, which store all historical trajectories. If an LBS monitors every query issued by a user,

1087-4089/12 $26.00 © 2012 IEEE
DOI 10.1109/I-SPAN.2012.35

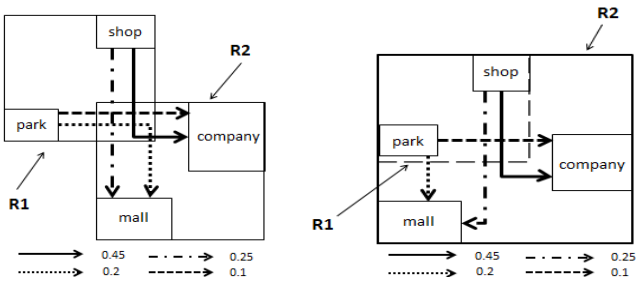


Figure 1. Examples of the most possible path for a user.

it is less likely to infer the actual user trajectory and as a result the user trajectory is indistinguishable from the $(r-1)$ trajectories in the long-term. Second, for k-anonymity, the principle is based on the traditional k-anonymity paradigm where each query location must contain k users including the user such that for each point query, the user is indistinguishable from $(k-1)$ users. However, k-anonymity has a serious problem in a high density area, since the range computed by the k-anonymity technique tends to be very small. Hence, it is easy to pin point where the user is located. Consequently, since we use the road network data, we consider the s-segment paradigm which requires that each query range must not only contain k users but also s road segments to avoid the k-anonymity problem. Finally, the fourth factor is time obfuscation. When a user moves toward his/her destination, the trusted anonymity server breaks the sequence of the query issuing time to confuse the LBS from knowing the user trajectory by randomly sending a query from a set of locations residing at the trajectories R.

The rest of this paper is organized as follows. In Section II, we highlight the related work. In Section III, we provide an overview of our system architecture and parameters. In Section 4, we propose our privacy techniques in detail for trajectory privacy. Our experimental results are shown in Section 5 and we conclude our work in Section 6.

II. Related Work

Previous techniques for location privacy mainly focused on snapshot queries. The most popular concept is to utilize k-anonymity to transform a point location into a cloaked region which makes the user indistinguishable from other $(k-1)$ users [2], [4], [5], [9], [11], [15]. To solve the k-anonymity problem, [12] and [14] adopted the real-world conditions to avoid cloaking an area into a small region from which the actual user location can be easily identified. For continue queries, Xu *et al.* [13] used $(k-1)$ historical trajectories to satisfy the trajectory-based k-anonymity paradigm. Since a trajectory is decomposed into a series of footprints, the proposed KAT technique cloaks the footprints on the k trajectories including the user trajectory and each of the footprints are in accordance with the chronological order. However, when a user moves on the cloaked path, the LBS can still easily identify the user's actual location, if there is no other user in that path. Although the user trajectory is under the k-anonymity protection, this work does not consider the ambient conditions which may raise a lot of privacy concerns. In [6], Kido *et al.* proposed two methods, moving in a neighborhood and moving in a limited neighborhood, to generate more reasonable dummy locations, enabling the trace of the user under the k-anonymity protection. In [7], Lei *et al.* utilized a rotation scheme to rotate a user trajectory and satisfy the distance deviation to make the user trajectory indistinguishable from the dummy trajectories. Furthermore, another k-intersected technique was proposed to decrease the probability of disclosing the user actual trajectory. The major issue of these dummy techniques is that when generating dummy locations or trajectories, they do not consider realistic locations. If a dummy location is generated on an unreasonable location, for example, on rivers or mountains, the LBS can easy filter out these locations, making the dummy location useless. In [1], Chow and Mokbel proposed the k-sharing region method; they believed that each continuous query location must contain $(k-1)$ identical users, such that the LBS cannot identify the user location by observing the entire continuous queries. The major issue of a k-sharing region is that as time goes by, users who are in the same cloaked area may move on different directions. As a result, the anonymity server generates a larger cloaked area that incurs performance degradation. In [8], Meyerowitz and Choudhury designed a prediction engine to predict the future possible paths according to the user past trajectories. On the prediction path, a trusted server sends queries to an LBS, and caches the results associated with the query locations. When a user approaches the location, the system restores the cached results for the user. In [8], the malicious LBS can easily identify the actual user trajectory when there is no other user on the predicted path. The prior work does not take the time factor into consideration. As a result, when continuous queries are issued in a time sequence, an LBS can track all issued queries to specular possible trajectories. In this paper, we consider this privacy concern and propose a new technique to solve the above-mentioned problems.

III. System Overview

Figure 2 shows the system architecture. Mobile users communicate with a trusted anonymity server over an encrypted communication tunnel. The trusted anonymity server plays a secure role for mobile users who transmit their messages to semi-honest LBSs. The anonymity server collects the user trajectory information and stores the trajectory data in a database which contains the historical trajectories of all users. When a user issues a continuous query with a pre-defined trajectory consisting of a starting and an ending point, it searches the database to satisfy the r-

anonymity policy. When a user travels on his/her trajectory, the anonymity server modifies each query location of the user into a cloaked region, changes the user id into a pseudo id which is encapsulated in the query message, and uses the time-obfuscated technique to send queries in a random sequence to confuse an LBS. When an LBS sends back the query results to the user, the trusted anonymity server filters the results to retrieve the query results to the user and caches these results for future usage, if the corresponding query location has not reached by the user. These techniques ensure that an anonymity server guarantees not only the privacy protection but also the correctness of queried results.

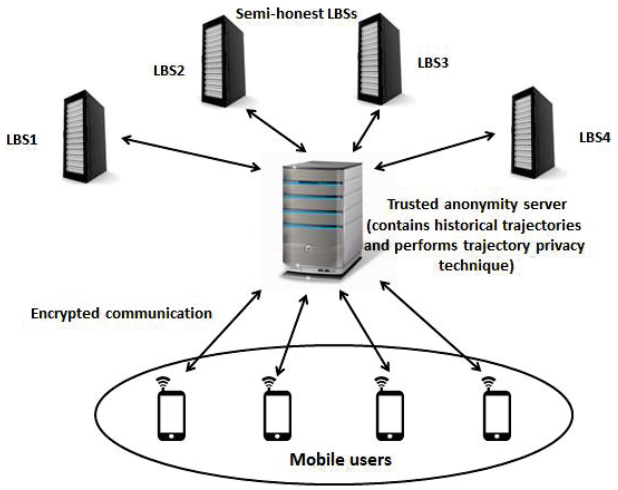


Figure 2. System architecture.

A. Definitions

The definitions and symbols we use throughout the following sections are defined as follows.

Definition 1: **A Trajectory Set** R:
R = {T_1, T_2,...,T_r}, where each T_i, $i \in 1...r$ is a trajectory composed by a set of footprints (i.e., a longitude-latitude position tracked by a GPS device on the way of a user moving toward the destination).
First, we define a set of trajectories R which consists of a user trajectory T_u and $(r-1)$ historical trajectories obtained from a database.

Definition 2: **A Trajectory** T_i:
$T_i = \{loc^i_{t_1}, loc^i_{t_2}, ..., loc^i_{t_n}\}$, where $loc^i_{t_j}$ is a footprint on trajectory T_i at time t_j.
Second, a trajectory consists of a set of footprints which are the positions tracked by a GPS device. To achieve time obfuscation, a query processor randomly selects a $loc^i_{t_j}$ on T_i to issue a point query at a random time t_j. In other words, the query processor does not periodically issue a query from a sequential location.

Definition 3: **A Query Message** m_i:
$m_i = (u_{id}, loc^i_{t_j}, k, s, resultTimeout, C)$, where u_{id} is a query user id. $loc^i_{t_j}$ is a query location on trajectory T_i at query issuing time t_j. (k,s) is a parameter set defined in a user privacy profile, which contains the k-anonymity and s-segment values. $resultTimeout$ is a time-out value for a valid query result. The user can set this value according to a query type and finally, C is the content of a query message. Third we define a query message as m_i, where i indicates a trajectory id in the trajectory set R. Before sending a query message m_i to an LBS, a trusted anonymity server modifies the m_i to an obfuscated query message m^*_i by modifying the u_{id} to $pseudo_{id}$ and $loc^i_{t_j}$ to $CloRange^i_{t_j}$.

Definition 4: **A Continuous Message Set** M:
$M = \{m^*_1, m^*_2, ..., m^*_z\}$, where z is the number of total queries received by an LBS.
Finally, we define a continuous message set M received by an LBS from the query positions on the r trajectories. Each query message m^*_i is issued from a query position (a footprint) on T_i, which is also randomly selected from the trajectory set R.

B. Privacy Factors

We explore four privacy factors in this paper: the r-anonymity, k-anonymity, s-segment and time-obfuscated techniques. The introduction to these factors are presented here.

The r-anonymity Paradigm When a user starts to communicate with an LBS, the r parameter is set on the trusted anonymity server. Next, the trusted server searches the historical trajectories in a database to find the nearest and similar $(r-1)$ trajectories to the user trajectory T_u to blur an actual user trajectory into r trajectories, which decrease the probability of linking the user trajectory T_u in the long-term. In addition, the trusted server randomly issues queries from the query locations on the trajectories R to avoid an LBS from knowing the user trajectory T_u by assembling each query location issued by user u.

The k-anonymity Paradigm The trusted anonymity server complies with the user privacy profile (k value) to generate a cloaked area which contains k users to achieve a protection. If a malicious LBS infers a user location, this technique successfully makes the user's identify indistinguishable from other $(k-1)$ users.

The s-segment Paradigm This factor is to used to solve the k-anonymity potential problem. As the user is in a high density area, using the k-anonymity principle to search for $(k-1)$ users often makes a cloaked range very small.

As a result, it is easy to expose the user actual location, even when the range contains k users. Thus, we combine the s-segment with k-anonymity paradigms which result in a cloaked region bounded with real world conditions. In summary, a trusted anonymity server first searches for k users in a cloaked region that contains at least s road segments.

The time-obfuscated Technique When a user starts to travel on his/her way, the trusted anonymity server starts to randomly send queries to an LBS. Unlike the previous privacy techniques, our server blurs the sequence of query issuing time and randomly chooses a query location from one trajectory in R. For example, when user u is located at query position $loc^u_{t_1}$, the anonymity server may issue a query from a position $loc^u_{t_9}$ or $loc^r_{t_5}$ from a trajectory $T_r \in R$ to confuse the LBS of knowing the current position of the user. This technique provides a strong privacy protection, prohibiting the LBS from disclosing a user starting location, destination and directions. Furthermore, the trusted server does not follow the chronological sequence of query issuing time to send the queries. We randomize each query issuing time to reduce the probability of reconstructing a user trajectory by putting all query locations together. When receiving a query from an anonymity server, the LBS cannot distinguish a query sent by the user from a query (from one of trajectories R other than T_u) generated by the anonymity server. By combining these two methods, we maximize the obfuscation level for a continuous query and meanwhile, increase the caching usability.

IV. Privacy Protection Algorithms

The main structure of our privacy technique is shown in Figure 3. We divide the structure into three parts. Firstly, for the CellMap component, we partitioning the map into grid cells with a fixed length and width, and store the number of segments on each corresponding cell. After partition the map, our privacy algorithm is implemented and the query location information is retrieved based on the grid indexing structure. Secondly, for r-anonymity, the details are introduced in Section IV-A. If less than $(r-1)$ trajectories are found, an anonymity server uses the virtual route planning system (e.g., Goolge map APIs) to generate the rest of the trajectories to satisfy the user privacy profile. Finally, the trusted anonymity server uses the trajectories R as an input data to perform a time-obfuscated algorithm and the details of the algorithm are introduced in Section IV-B.

A. The r-anonymity Algorithm

The pseudo code of the r-anonymity paradigm is shown in Algorithm 1. The inputs include a r value for the user privacy profile, historical trajectory sets, and a user trajectory T_u. We use three arrays as the data structure to represent a trajectory: (1) a user list consisting a list of registered users

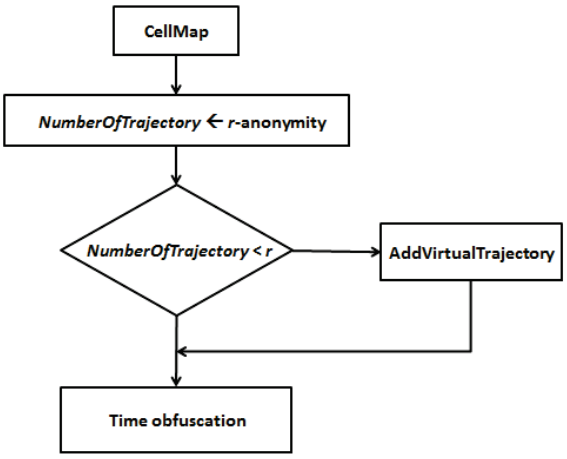


Figure 3. Main structure.

in the system, (2) a user trajectory list, and (3) a cell-based footprint list. For the $UserTrajectory$ structure, we use an array to represent it. The output of this algorithm is a two-dimensional array (termed r-trajectory array) repositing the r trajectories. The first and second dimension store the trajectory index and a footprint list, respectively. Finally, we return the total number of trajectories to instruct the server to determine whether or not to perform the virtual route planning mechanism to add more virtual trajectories.

In Algorithm 1, beginning with the r-anonymity process, we assign the user trajectory T_u into the r-trajectory array and compute the length of the user trajectory for bounding the search space for r trajectories. Then, the algorithm searches a database D for the historical trajectories by the following steps. First, in Lines 8-10, we check the length. If the historical trajectories are shorter than $(UserLength * MIN\_LENGTH)$ or longer than $(UserLength * MAX\_LENGTH)$, they are useless in helping blur the user trajectory. For example, if the trajectory only has a few or too many footprints, an adversary may link these r trajectories and reveal the r trajectories since an extremely short-length or long-length trajectory is not likely to be issued by a user in reality, such that an adversary can easily identify this synthetic trajectory. Thus, in our approach, we aim to generate $(r-1)$ trajectories with a similar length to that of the user trajectory. Second, in Lines 11-13, the overlapping trajectories are checked. After filtering out the trajectories with unreasonable length with respect to the length of the user trajectory, we check T_u and the trajectories which already have been assigned to the r-trajectory array to avoid duplicate trajectories. For example, if n trajectories in the r-trajectory array are duplicate paths after searching for the r trajectories, the system uses the locations on the duplicate trajectories to send queries. In such a case, the adversary may link these queries and

reconstruct the whole possible trajectories. Eventually, an adversary may gather $(r-n)$ trajectories, breaking the user privacy profile. Therefore, we set the minimum percentage for overlapping trajectories to generate the r trajectories. Third, in Line 14, we computed the distance variance to search for the nearest trajectory to the user trajectory T_u. This mechanism helps the final $(r-1)$ trajectories close to the user trajectory such that the r trajectories occupy fewer cells. Hence, the maintenance overhead for the grid indexing is reduced.

Algorithm 1 *r-anonymity*

1: Input: (r,D, $UserTrajectory$)
2: Output: (r-$Trajectory$, $NumberOfTrajectory$)
3: Assign $UserTrajectory$ to r-$Trajectory$
4: $UserLength$ = *LengthOfTrajectory*($UserTrajectory$)
5: **for** (i = 0 TO r) **do**
6: **for** (j = 0 TO $totalUser$) **do**
7: **for** (k = 0 TO $|D|$) **do**
8: $TrajectoryLength$=*LengthOfTrajectory*($D.T_k$)
9: /*The trajectory must have a proper length to avoid retrieving few footprints on a trajectory.*/
10: **if** ($TrajectoryLength > (UserLength * MIN\_LENGTH)$ and $TrajectoryLength < (UserLength * MAX\_LENGTH)$) **then**
11: $Overlap$ = *ComputeOverlap*($r-Trajectory$, $D.T_k$)
12: /*If a historical trajectory overlaps too much with the user trajectory, it reduces the blurring effect.*/
13: **if** ($Overlap < MAX\_OVERLAP$) **then**
14: $distVarance$ = *DistVarience*($UserTrajectory$, $D.T_k$)
15: **end if**
16: **end if**
17: **end for**
18: **end for**
19: /*Find the smallest distance variance for obtaining the r trajectories.*/
20: r-$Trajectory$ = *FindMinRVar*($distVarance$, $D.T_k$)
21: **end for**
22: return $NumberOfTrajectory$

B. The time-obfuscated Algorithm

The pseudo code of the time-obfuscated algorithm is shown in Algorithm 2. The inputs include two user privacy profile values k and s, a user trajectory T_u, r trajectories and cache result time-out time. When a user starts to travel toward the distinction, an anonymity server adopts the time-obfuscated technique to randomly send queries, including a query time, at which the user does not issue a query. This method is performed until the user reaches the destination. At every random query time, the server first checks whether there are any time-out results cached in the server in Line 4. Since the anonymity server randomly selects query locations on the r trajectories to issue queries, any results returned from the LBS is associated with an expiration time. When the results expire, they are useless to the user. We need to remove these results and re-issue queries from these locations as the system proceeds. Lines 5-15 check a query time to determine whether or not to issue a query for the user. If the system determines to send a query, it selects a user footprint and randomly chooses a query index. At any query issuing time, we aim to send more than one queries to an LBS, because it causes confusion for the LBS to identify the user. Furthermore, if the system determines not to send a query, the algorithm randomly decides whether to use a user footprint as a query location.

We particularly added a random mechanism to determine a user footprint as a query location to increase the cache usability. If the user has not reached a location $loc^u_{t_j}$, but the anonymity server may still use $loc^u_{t_j}$ as a query location to send query in advance. When the user arrives $loc^u_{t_j}$, the anonymity server reuses the cached results to retrieve the query result for the user and therefore, the total number of queries is reduced. This function improves the server efficiency and enhances the user privacy level. Next, an anonymity server starts to send queries and checks the index for every query of the user. If a query belongs to the user, the server assigns the location on the user trajectory as a query location, and if the query is not sent by the user, it randomly chooses a location from r trajectories. After that, the anonymity server uses the query location to search for $(k-1)$ users in Line 24, because a location is represented by a cell, the anonymity server just searches the users who are within the cell. If the cell does not contain enough $(k-1)$ users, the anonymity server searches the neighboring cells until it satisfies the k-anonymity policy. The server then sorts the cells according to the number of users in descending order and start processing the cell with the highest number of users until k users are completely gathered. If there are less than k users, it outwardly expands the search range to search more cells until the number of user reaches k. After satisfying the k-anonymity policy, the anonymity server forms these cells into a range and checks this range to see whether it contains s road segments or not. If not, it uses the same way as we obtain for k users in Line 32, to search on the neighboring cells until there are s segments. After satisfying the user privacy profile, the server uses this cloaked range as a query location and changes a user id into a pseudo id to send the query to an LBS. When an anonymity server receives the results from an LBS, it caches the results and assigns these cells a time-out stamp in Line 36 to keep tracking of the up-to-date search results.

V. Experiments

We use a real dataset downloaded from CRAWDAD [3] as our simulation data and it contains mobility traces of taxi cabs in San Francisco, USA. This dataset contains GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area, and we use 200 taxis as our simulation data. Since some taxis travel outside the San Francisco Bay Area, we filter out these GPS coordinates beyond the bounding box represented by four latitude-longitude points, (37.81, -122.53), (37.81, -122.35), (37.7, -122.53), (37.7, -122.35) as shown in Figure 4.

In Table 1, we capture 8560 road segments from a Google map, and we set five different sizes to partition the map, the length of a cell is set to 100m, 200m, 300m, 400m and

Algorithm 2 *The time-obfuscated algorithm*

Input: (k,s, $UserTrajectory$, r-$Trajectory$, $resultTimeout$)
$issueTime$ = $CurTime$ + $random(MAX\_QUERY\_TIME)$
while ($CurTime$ == $issueTime$) **do**
  RemoveTimoutQuery($resultTimeout$)
  if (*SearchQueryLoc*($UserTrajectory$) returns true) **then**
    /*Randomly assign a query index within MAX_QUERY_NUM.*/
    $queryIndex$ = $RanQueryIndex(MAX\_QUERY\_NUM)$
    $UserQueryCell$ = $UserTrajectory$
  else
    $isQuery$ = *RanDecideUserQuery*($UserTrajectory$)
    if ($isQuery$ == true) **then**
      $queryIndex$ = $RanQueryIndex(MAX\_QUERY\_NUM)$
      $UserQueryCell$ = *RanChooseLoc*($UserTrajectory$)
    end if
  end if
  /*Sending the queries*/
  for (j = 0 TO MAX_QUERY_NUM STEP 1) **do**
    if ($queryIndex$ == j) **then**
      $QueryCell$ = $UserQueryCell$
    else
      $QueryCell$ = *RanChooseCell*(r-$Trajectory$)
    end if
    /*Counting then number of users located at a QueryCell location.*/
    $userNum$ = *ComputeUser*($QueryCell$)
    if ($userNum$ ¡ $k-1$) **then**
      /*Searching on the neighbor cells until the number of users reaches k.*/
      $CloakRange$ = *SearchNeighborForK*($QueryCell$)
    end if
    /*Checking the number of road segments in the query range.*/
    Seg = *CheckSegmentNum*($CloakRange$)
    if (Seg ¡ s) **then**
      SearchNeighborForS($CloakRange$)
    end if
    SendQuery($CloakRange$)
    /*Assigning a time-out time to the cloaked region.*/
    AssignTimeout($CloakRange$, $ResultLiveTime$)
  end for
  Increment $CurTime$
  $issueTime$ = $CurTime$ + $random(MAX\_QUERY\_TIME)$
end while

Figure 4. Filter area.

500m. Therefore we obtain 19276 cells for a 100m × 100m grid, 4819 cells for a 200m × 200m, 2173 cells for a 300m × 300m, 1240 cells for a 400m × 400m, and 660 cells for a 500m ×500m grid. Then we set the user privacy profile as follows. The r value is set to 5, 10, 15, and 20; the k value is set to 5, 10, 15, and 20; the s value is set to 10, 20, and 30. We use various parameters to conduct our simulations and for each simulation, there are ten different users at each experiment. Finally, we set the time-out time to 1 minute, 2 minutes, 3 minutes, and 4 minutes for each query result. As the descriptions are presented in Section IV-B, when cached results are expired, the server removes the old results and re-issues queries from the locations which contain the expired results.

Parameters	
Side length of cell	100m, 200m, 300m, 400m, 500m
Number of cells	19276 (100m), 4819 (200m), 2173 (300m), 1240 (400m), 660 (500m)
r value	5, 10, 15, 20
k value	5, 10, 15, 20
s value	10, 20, 30
Result time-out time	1, 2, 3, 4 minutes

Table 1: Experimental parameter settings.

A. The r-anonymity Paradigm

In this experiment, we show the cell distribution of the r trajectories before and after the computations in Figure 5. The black arrow line is used to represent a user trajectory and direction. We use the cell size of 200m × 200m as a partition size and set k to 10 and s to 20 for the privacy profile.

In Figures 5 (a), (b), (c), and (d), we can observe that as r increases, the distribution of cells become a broader area (the shaded area), which is more likely to conceal the actual user trajectory from an LBS. After the query processing is completed, an adversary may have difficulty linking the entire queries to reconstruct the user trajectory. However, as we can see in Figures 5 (c) and (d), the user trajectory is hidden in these cells and it is almost impossible to reveal the original cells blurred by the r trajectories. These results prove that our r-anonymity guarantees to preserve user trajectory privacy in a long-term.

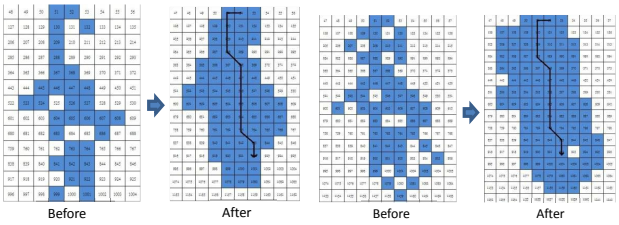
B. The k-anonymity and s-segment Paradigms

We use three results to prove that our privacy technique guarantees a strong privacy protection while considering a user privacy profile.

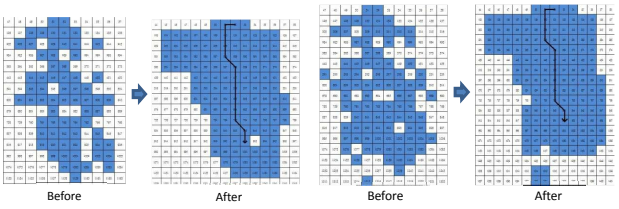
1) Cloaked Regions: We use km^2 to evaluate cloaked regions in Figure 6, where we vary the cell size to obtain different cloaked regions. We set s to 10, 20 and 30 and vary k from 5 to 20 to combine with different k of 5, 10, 15, and 20. We fix the number of users to 10 when running each set of simulations.

As we can see in Figure 6, when k increases, the average km^2 region becomes broader. However, the impact of s is not significant. In addition, when partitioning the map with a small cell size, the system achieves a better privacy protection. In general, the average cloaked region is under 1 km^2 and a strong privacy protection is still maintained.

2) The k-anonymity Paradigm: As shown in Figure 7, we use different cell size and s to verify the number of users in a cloaked region which contains at least k users specified in the privacy profile.

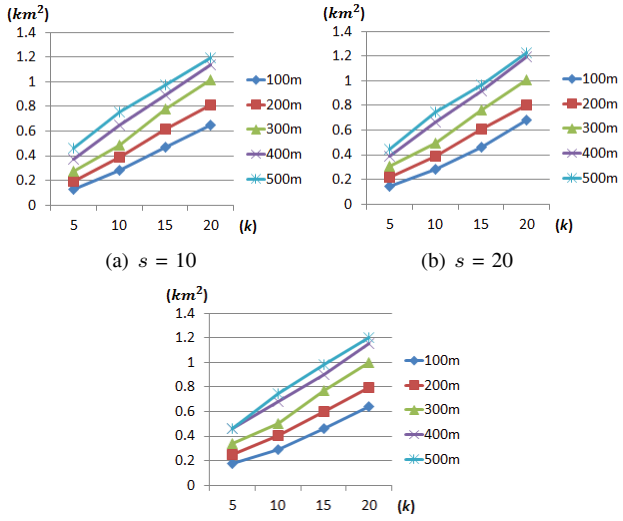


(a) r = 5 (b) r = 10



(c) r = 15 (d) r =20

Figure 5. The r-anonymity schematic diagram.



(a) s = 10 (b) s = 20

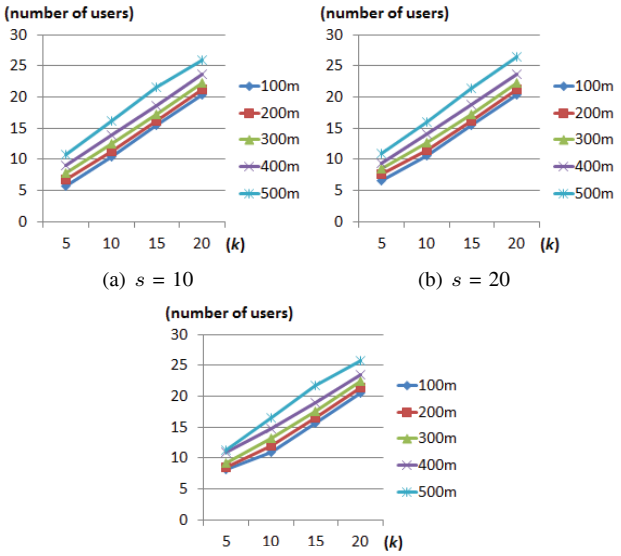
(c) s = 30

Figure 6. Cloaked regions.

As we can see here, the average number of users exceeds k. In case of a cell size with 500m $\times$ 500m, the number of users is almost five times more than k. From these results, we can see that our approach satisfies the k-anonymity policy to support a strong privacy protection.

3) The s-segment Paradigm: We show the average number of road segments found in the query location. In Figure 8, we vary the cell size and use different k to verify the impact of road segments.

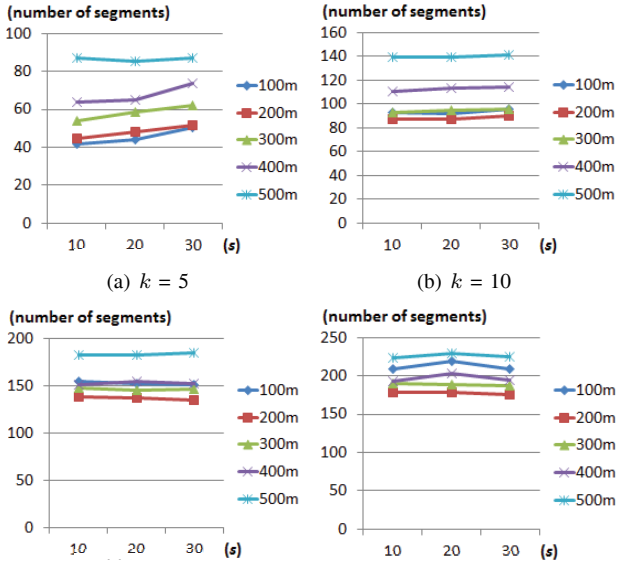
As we can see here, when the s increases, the number of road segments remains virtually unchanged. When a query is issued, our privacy algorithm obtains k users first and subsequently, continues the process until there are enough s road segments. The results are expected since a cell with a large size is very likely to cover more road segments.



(a) s = 10 (b) s = 20

(c) s =30

Figure 7. Average users in a cloaked region.



(a) k = 5 (b) k = 10

(c) k = 15 (d) k = 20

Figure 8. Average number of segments in a cloaked region.

C. Number of Duplicate Queries

Figure 9 illustrates the number of duplicate queries sent by the server using our approach. If the duplicate queries are too many, the performance of the anonymity server is degraded. In our experiments, to test the impact of duplicate queries, the parameters r, k and s are set to 10, 10, and 20, respectively and we vary the cell size and expiration time. The number of users is set to 10, and we compute the average number of queries. When the time-out time is extended, the number of duplicate queries is gradually reduced. Note that when the cell size is set under 300m

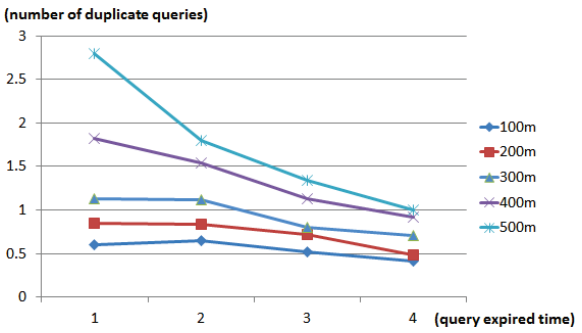


Figure 9. Duplicate query number.

$\times$ 300m, the number of total queries is only twice more. Because when a cell size is set to a higher number, the total number of cells overlapped with the r trajectories becomes fewer, such that the anonymity server can only use fewer locations to randomly send queries. However, these results prove that our time-obfuscated technique does not increase overheads dramatically on the anonymity server and achieves a better privacy protection.

VI. Conclusions

We propose a novel technique for a trajectory privacy protection, which considers a long-term disclosure issue as well as the real-time ambient conditions. We allow the user to specify their privacy profile and our method integrates the r-anonymity, k-anonymity and s-segment paradigms. We also introduce a time-obfuscated technique to increase the level of trajectory privacy protection dramatically. As the experimental results have shown, our privacy technique protects the user trajectory, preventing the LBS from reconstructing a user actual trajectory and the direction. Our method guarantees to satisfy the user's privacy profile for providing a stronger privacy protection.

Acknowledgments

This research has been funded in part by the National Science Council under the Grants NSC 100-2221-E-194-027-MY3 and NSC 101-2221-E-194-054.

References

[1] C.-Y. Chow and M. F. Mokbel. Enabling private continuous queries for revealed user locations. In *Proceedings of the 10th international conference on Advances in spatial and temporal databases*, SSTD'07, pages 258–273, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, GIS '06, pages 171–178, New York, NY, USA, 2006. ACM.

[3] CRAWDAD. http://www.crawdad.org/index.php.

[4] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, Jan. 2008.

[5] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, MobiSys '03, pages 31–42, New York, NY, USA, 2003. ACM.

[6] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *In: ICPS. (2005*, pages 88–97, 2005.

[7] P.-R. Lei, W.-C. Peng, I.-J. Su, and C.-P. Chang. Dummy-based schemes for protecting movement trajectories. *J. Inf. Sci. Eng.*, 28(2):335–350, 2012.

[8] J. T. Meyerowitz and R. R. Choudhury. Realtime location privacy via mobility prediction: creating confusion at crossroads. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, HotMobile '09, pages 2:1–2:6, New York, NY, USA, 2009. ACM.

[9] 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 '06, pages 763–774. VLDB Endowment, 2006.

[10] OnStar. http://www.onstar.com.

[11] X. Pan, J. Xu, and X. Meng. Protecting location privacy against location-dependent attacks in mobile services. *IEEE Transactions on Knowledge and Data Engineering*, 24:1506–1519, 2012.

[12] K. Wei-Shinn, C. Yu, and Z. Roger. Privacy protected spatial query processing for advanced location based services. *Wireless personal communications (Dordrecht. Online)*, 51(1):53–65, 2009. eng.

[13] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location-based services. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 547 –555, april 2008.

[14] M. Xue, P. Kalnis, and H. K. Pung. Location diversity: Enhanced privacy protection in location based services. In *Proceedings of the 4th International Symposium on Location and Context Awareness*, LoCA '09, pages 70–87, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] L. Yao, C. Lin, X. Kong, F. Xia, and G. Wu. A clustering-based location privacy protection scheme for pervasive computing. *CoRR*, abs/1011.3098, 2010.