Fingerprinting Mobile User Positions in Sensor Networks

[1,2]Mo Li, [1]Xiaoye Jiang, and [1]Leonidas Guibas
[1]*Computer Science Department, Stanford University, CA, USA*
[2]*CSE Department, HKUST, Hong Kong*
limo@cse.ust.hk, {xiaoyej, guibas}@stanford.edu

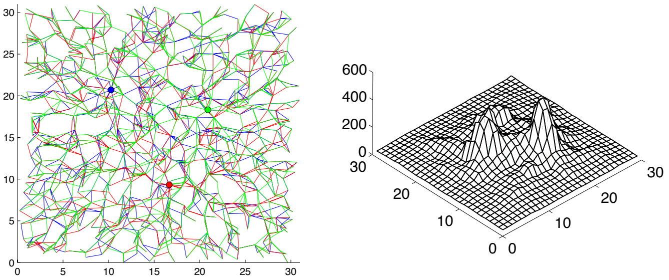
***Abstract* - We demonstrate that the network flux over the sensor network provides us fingerprint information about the mobile users within the field. Such information is exoteric in the physical space and easy to access through passive sniffing. We present a theoretical model to abstract the network flux according to the statuses of mobile users. We fit the theoretical model with the network flux measurements through Non-linear Least Squares (NLS) and develop an algorithm that iteratively approaches the NLS solution by Sequential Monte Carlo Estimation. With sparse measurements of the flux information at individual sensor nodes, we are able to identify the mobile users within the network and instantly track their movements without breaking into the details of the communicational packets. A particular advantage of this approach is that compared to the vast information we can reveal the required knowledge is extremely cheap. As all fingerprint information comes from the network flux that is public under current wireless communication medium, our study indicates that most of existing systems are vulnerable in protecting the privacy of mobile users.**

Keywords—sensor networks; network flux; mobile user; fingerprint

I. INTRODUCTION

Recent advances in wireless sensor network (WSN) technologies envision more pervasive usage of the sensor network where the human beings are deeply interacting with the cyber-physical environment. In addition to the traditional paradigm of data collection from remote sensor networks, people may coexist in the same physical space of interest with the sensor network infrastructures. Equipped with 802.15.4 compatible communicating devices, each user is able to move around within the sensor network and directly communicate with nearby sensors, capable of pervasive access to the instant data over the entire field.

In such a pervasive context of data access, the deployed infrastructural sensor network is capable of simultaneously supporting multiple mobile users and providing them with field data in an anyone-anywhere-anytime manner. There have been substantial applications based on this data access mechanism, from ubiquitous data acquisition to human navigation, and etc [11, 13]. The mobile users access the network at different locations and acquire network-wide data instantly through intermediate nodes.



(a) **(b)**

Figure 1. The network flux with three mobile users. (a) The data collection trees; (b) The network flux pattern.

In this paper, however, we demonstrate that such a working paradigm suffers from a potential risk of leaking the location privacy of users. With alarming ease, a malicious entity can track the every move of mobile users only from passively sniffing the network traffic flux at a sparse set of points. They do not even need to break into the content of data packets.

The mobile users access the network at different locations and produce their own traffic flows respectively across the network. In most of existing works, a data collection tree is built for each mobile user and network-wide data are delivered by intermediate sensor nodes along the tree [10, 14]. The produced traffic flows of different mobile users add upon each other at intermediate nodes and the traffic amounts cumulate. If we summarize the traffic flux distributed over the network we get a flux pattern of particular shape. Figure 1 depicts the network flux pattern where there are 3 mobile users collecting data from the network. Figure 1 (a) presents the three mobile users and their data collection trees built across the network and Figure 1 (b) depicts the network flux pattern introduced by the mobile users. Indeed, the pattern of the network flux is related to the statuses of mobile users. It digests the information including the number of mobile users, their locations, their traffic stretches, and etc. Thus by exploring the traffic pattern over the network, we are able to build a mapping between the instant distribution of mobile users and the observed network flux.

We build a parameterized model to abstract the network flux with different situations of mobile users. By fitting the theoretical model to the measurements on real network flux, we are able to gradually identify the locations of mobile users distributed over the field. While gathering the flux information over the entire network might be of heavy overhead, we show that even with sparse measurements of the flux at a small set of individual sensor nodes we are still able to fingerprint the mobile users through parameter fitting. We further develop an algorithm that iteratively approaches the movements of mobile users by Sequential Monte Carlo Estimation technique. Our algorithm takes a time series of flux measurements as inputs. At each time instance, the possible locations of mobile users are predicted by a set of weighted samples that approximate their posterior distribution. The samples are filtered and updated according to the NLS fitting result and new predictions are drawn from them. As more flux measurements are cumulated, our algorithm converges to the moving trajectories of mobile users and approximates their locations with high accuracy.

A particular advantage of this approach is that compared to the vast information we can reveal, the required knowledge is extremely cheap. Only sparse knowledge of the network flux is enough for the entire calculation. As a matter of fact, due to the broadcast nature of the wireless communication medium, such information is easy to access through passive sniffing. We only grasp the amount of traffic flux at each individual node instead of taking out the concrete flow information, i.e., we do not need to look into the transmitted packets which are much more expensive and often difficult to access. As a direct result, we demonstrate that most existing systems are vulnerable in protecting the privacy of mobile users. The malicious entities can easily identify the mobile users within the network and instantly track their movements without breaking into the details of the network communications. Any cryptographic mechanisms cannot protect the privacy of mobile users from revealing their movements in the network flux information.

The remainder of the paper is organized as follows. Section 2 discusses existing work related to this study. Section 3 describes the main design rationale. In Section 4, we give detailed descriptions on how we fingerprint the mobile users with sparse samplings of network flux. In Section 5, we validate our design with extensive simulations. Finally, we conclude this work in Section 6.

II. Related Work

Knowing accurate locations of interesting objects or people is of essential importance for many pervasive applications. Initial attempts of the research community include LANDMARC [17], RADAR [1], Cricket [20], and etc. There have been also many approaches proposed for locating and tracking objects within the sensor network.

Some approaches aim to automatically determine sensor locations once upon the network is deployed, referred to as self-localization. A general overview of the state-of-the-art schemes is available in [6]. Basically, those approaches rely on a set of beacon nodes with known locations. Other nodes measure physical distances through ranging techniques or virtual distances within the network and compute their own locations based on the beacon nodes and the distance measurements. Various ranging techniques have been applied for distance measurement, such as Time of Arrival (TOA) [25], Time Difference of Arrival (TDOA) [21], Radio Signal Strength (RSS) [1], Angle of Arrival (AOA) [18], and etc. Many techniques have also been developed to compute the locations with such measurements, from global embedding and optimization to sequential triangulation or sweeping [5, 7, 12, 16]. For self-localization, both the network infrastructure and the nodes to be determined are cooperative, i.e., we can easily access information exchanged among nodes within the network, such as location beacons, distance measurements, network structures and so on, which provides us adequate knowledge for location calculation.

Some approaches aim to remotely locate external objects in the sensor network field, referred to as remote localization. Techniques like ultrasound, infrared, or RF Doppler effect based detection methods are developed for accurate object detection [9, 20, 24]. By sequentially computing the instant locations of remote objects it is possible to track their movement trajectories. Instead of directly estimating the static locations of the object at discrete time instances, constrained nonlinear least squares (CNLS) and extended Kalman filter (EKF) are usually applied to establish the motion model of objects and achieve higher accuracy [9, 23]. Different from self-localization, in remote localization and tracking applications the target object may not always be cooperative, e.g., the intruder detection, wildlife tracking, and etc [22, 26]. Nevertheless, the sensor network system itself is open, providing us all operational information that is needed.

Different from all existing studies, in this work we demonstrate that even when both the moving entities and the sensor network infrastructures are non-cooperative, we can still identify the mobile users with minimum information that is difficult to secure. Our research result implies that there exists potential threat towards protecting user privacy in existing sensor network systems.

The problem of disclosing user privacy in wireless network context has recently drawn the concern of research community. There have been studies showing that the location privacy could be vulnerable with the “broadcast” wireless communication channels [2, 4, 19]. They demonstrate that the adversaries are able to acquire user locations with wireless fingerprint information that can be obtained through direct or indirect access to the inbound and outbound traffic nearby the user. Most existing studies require direct access to the data packets or heavy monitoring of the traffic flows to obtain necessary fingerprint information. In this work, however, we show that a sparse sampling on the amount of traffic flux in the field suffices to reveal fair amount of location privacy of mobile users, which is much cheaper and easier for the malicious entities to launch.

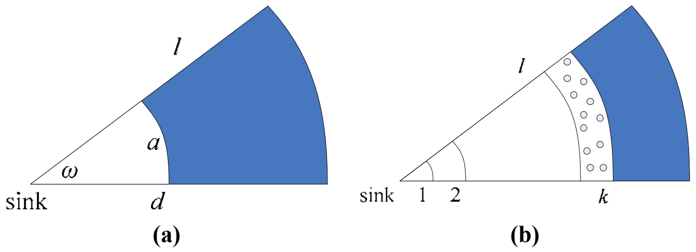


Figure 2. Illustration of the network flux model. (a) Continuous case; (b) Discrete case.

III. Design Rationale

Our goal is to solely utilize network flux information to fingerprint the mobile users within the sensor network field. In this section, we first formalize the problem we are studying, including the application scenario, design objective, assumptions, and etc. We then develop a parameterized model to predict the network flux over the field. We introduce our basic design rationale of locating mobile users through briefing the network flux.

A. Problem Statement

We consider the scenario where multiple mobile users move around within the sensor network field, collecting the sensory data from the network.

Let the number of mobile users be K. Each mobile user repeatedly collects the updated data from the network at its own will. The data collection of each user happens at different time and different places. For any mobile user i, there exists a time series of data collections [t^i_1, t^i_2, ..., t^i_{ki}] while the corresponding positions are [p^i_1, p^i_2, ..., p^i_{ki}]. Different users may have different time series of data collections independent of each other. Our goal is to track those mobile users, i.e., to figure out the location instances of each mobile user $\{[p^i_1, p^i_2, ..., p^i_{ki}] \mid 1\leq i\leq K\}$.

Towards such a goal, what we assume available are the instant measurements of the traffic flux over the network at each time window ΔT. The time window ΔT determines the measurement granularity. When $\Delta T\rightarrow 0$, we get ever more delicate observation of the network flux. In practice, ΔT is limited by the inherent duration of wireless transmissions, synchronization among different observers, and etc. Nevertheless, with current technologies, ΔT can be bounded at the "seconds" level, leading to minor observation error compared with the intrinsic system error brought by the discrete position estimations with "minutes" intervals. Within each time window, different mobile users may or may not happen to initiate the data collection. In a more general way as adopted in most existing works, when one mobile user wants to collect the data from the network, it builds a data collecting tree that roots at the sink and spans the network. Different mobile users may have different traffic stretches, i.e., they collect different proportions of data from each node due to their interests at different environment aspects. The measured network flux at each time window is the sum-up of the traffic F_i initiated by each mobile sink. At each node, we can measure the cumulated traffic flux:

$$F = \sum_{i=1}^{K} F_i$$

However, we cannot exactly separate each share of the flux amount introduced by each mobile user. Instead, we develop a mathematical model to fit the mobile user statuses according to such combined fingerprint flux information.

B. Network Flux Model

In this section, we study how the network flux is composed when the mobile user absorbs data from the network-wide data collecting tree. We accordingly build a network flux model to approximate the amount of data flux at each node.

Note that the data flux at each intermediate node is the cumulated amount of data it generates and relays, including the data generated at all successor nodes on the subtree it roots. We first consider a continuous scenario where sensor nodes are deployed over the field with infinite density. Figure 2 (a) depicts a sector-like region of angle ω and radius l originated at the user. We assume that each point within the sector-like region generates a unit of data and the traffic stretch is s for each unit area. For the arc a which is d distant from the sink, all data generated at points beyond a (in the blue area) pass the arc. Let the average traffic flux at each point on arc a be F_a. We have the entire amount of data delivered across a:

$$M_a = \int_{\theta=0}^{\omega}\int_{d}^{l} s\cdot r dr d\theta = \int_{a} F_a d(x,y) \tag{3.1}$$

From Equation 3.1, we get $F_a = s(l^2\text{-}d^2)/2d$, which is independent of the angle ω. We let $\omega\rightarrow 0$ and obtain that the flux at each intermediate point F_d is determined by the distance d from the sink to that point and the distance l from the sink to the network boundary along the direction of that point.

$$F_d = s(l^2\text{-}d^2)/2d \tag{3.2}$$

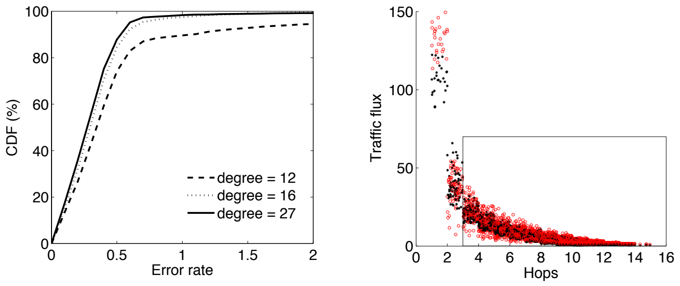
Formula 3.2 models the traffic flux for the ideal network of infinite node density. For a more practical flux model, we further generalize our analysis for the discrete networks. Figure 2 (b) illustrates how the data flux concentrates at the k-hop away nodes from the user. All k-hop nodes reside inside the strip area k hop distant from the user and all nodes beyond k hop away from the user (in the blue area) have their data amount relayed by those k-hop nodes. Let the flux at each k-hop node be F_k. We have the entire amount of data transmitted through those k-hop nodes is:

$$M_k = \frac{\omega(k^2r^2-(k-1)^2r^2)}{2}\cdot\rho\cdot F_k = \frac{\omega(l^2-(k-1)^2r^2)}{2}\cdot\rho\cdot s \tag{3.3}$$

where r is the average distance of each hop, ρ is the node density, and s is the traffic stretch of the current sink.

From Equation 3.3, we get $F_k = s(l^2\text{-}(k\text{-}1)^2r^2)/(2k\text{-}1)r^2$. We can reformulate it and approximate F_k with items of real distance variables:

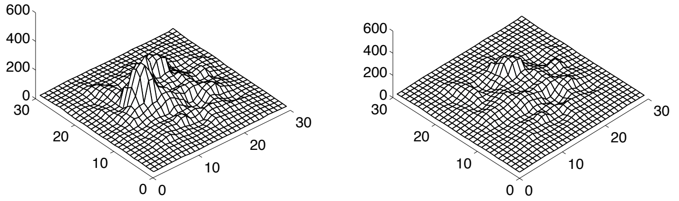
$$F_d \approx s(l^2\text{-}d^2)/2dr \tag{3.4}$$



(a)

(b)

Figure 3. Statistical results. (a) CDF of the approximation error rate; (b) Flux measurement v.s. approximation.



(a)

(b)

Figure 4. Briefing the network flux.

Indeed, Formula 3.4 is a consistent representation of Formula 3.2 in the discrete setting with a division factor of the average hop distance *r*. According to Formula 3.4, the position of the mobile user determines the parameter *l* and *d*, thus affecting the traffic flux at intermediate nodes. Using Formula 3.4, we are able to approximate the traffic flux at any position for general discrete networks. On the other hand, if we have the measurements of the traffic flux at each node, Formula 3.4 allows us to identify the location of the mobile user with a parameter fitting. Indeed, if we average the amount of flux within the neighborhood of an intermediate node, we are able to get a smoother map of the network flux and better approximation accuracy by mitigating the randomness of routing tree construction.

To examine the approximation accuracy of this model we simulate uniform random networks of 2500 nodes on a square field. Figure 3 presents the statistical results. Figure 3 (a) plots the cumulative distribution of the approximation error with different network densities. According to the statistics, the traffic flux of most nodes (80%+) can be well approximated with less than 0.4 error rate. As the node density of the network increases, the error rate can be further reduced. Figure 3 (b) plots the concrete traffic flux measurement (red dots) v.s. the approximated flux amount (black dots) according to our model. In this experiment, the average network degree is set to 12. Indeed, we find that the approximation error decreases with the network hops between the sink and the estimated nodes. If we focus on those nodes 3 hops away from the sink (denoted in the box in Figure 3(b)), we can get much lower approximation error and still preserve more than 70% energy of the network flux. This allows us to identify the position of the mobile sink more accurately with those nodes.

C. Briefing the Network Flux

According to the analysis in the previous section, with the summary of traffic flux over the network we are able to identify the location of the mobile user by simply extracting the point of traffic concentration. The problem, however, becomes a bit more difficult when there are multiple mobile users initiating data collection at the same time. As Figure 1 demonstrates, there are three mobile users collecting network data with three different data collecting trees and their traffics cumulate at intermediate nodes. Under such circumstances, simply detecting the traffic peaks is not effective any more. We cannot distinguish traffics of different mobile users and the traffic flux of one mobile user may heavily influence the observation on other mobile users, especially when different mobile users have different traffic stretches.

Against such a problem, we use a recursive method to brief the observed network flux with our theoretical model. We identify the positions of mobile users in multiple rounds. In each round, we detect the global traffic peak and accordingly identify the position of a mobile user. We then estimate the traffic stretch of the mobile user with the peak traffic. From the theoretical model, we are thus able to approximate the network traffic flux associated with the current mobile user. We then subtract its corresponding amount of traffic from the original network flux, facilitating later detection of other mobile users. By such a method, we always get reduced map of network flux and are able to identify the mobile user of dominating traffic at each round. In Figure 4, we demonstrate how we apply such a method to brief the network flux of the example shown in Figure 1. Figure 4 (a) depicts the reduced network flux map after one mobile user is identified. Figure 4 (b) depicts the map after two mobile users have been identified. We find that our network flux model well approximates the real observations and such a recursive method accurately identifies the distribution of mobile users despite that their traffics mix with each other. On the other hand, this method requires the flux information over the network to capture those traffic peaks. Such a requirement leads to expensive overhead, i.e., sniffing all the nodes within the entire network. In next section, however, we show that we can fingerprint the mobile users with only sparse samplings of the network flux, largely reducing the overhead.

IV. Fingerprinting with Sparse Samplings

Instead of acquiring the traffic flux information over the entire network, we can merely use sparse samplings on a small portion of nodes to get adequate fingerprint information. We do a parameter fitting on our theoretical flux model according to the node flux samplings over the network such that we can find the best possible distribution of mobile users. We further develop an algorithm that iteratively approaches the mobile sink movements by Sequential Monte Carlo Estimation technique.

A. NLS Parameter Fitting

With only sparse samplings from a small portion of nodes, we are not able to directly map out those traffic peaks of mobile user origins. Instead, we do parameter fitting on our theoretical flux model such that the flux measurements can be the best fit.

Assume that we have the flux samplings at n nodes. From the theoretical model indicated by Formula 3.4, we can estimate the flux vector F at sampling nodes and compare with the real measurement vector F'. The best parameter fitting corresponds to a Non-linear Least Squares (NLS) optimization problem, which will minimize the following objective function:

$$\min . \left\| F - F' \right\| \tag{4.1}$$

$$\begin{cases} F_i = \sum_{j=1}^{K} \frac{s_j}{r} \cdot \frac{(l_{i,j}^2 - d_{i,j}^2)}{2d_{i,j}}, (i = 1, 2, ..., n) \\ l_{i,j} = l(x_i, y_i, x_j, y_j) \\ d_{i,j} = d(x_i, y_i, x_j, y_j) \end{cases}$$

Here, F_i describes the estimated flux amount at the i-th node according to the flux model, where the traffic of K mobile users cumulates. The estimated value F_i is determined by the positions of mobile users (x_j, y_j), and their traffic stretches s_j. We try to fix such parameters as the solution $\in \Re^{3K}$ for this optimization problem. Indeed, the number of mobile users K is not necessarily preknown. For the cases where we do not know the exact number of mobile users in the field, we can conservatively choose a K large enough, and after the optimization process the K coordinates will converge at the actual positions of mobile users. There is an unknown constant r in the function which measures the average distance of each communication hop. In practice, r is limited by the maximum communication radius R, but is different under different network densities. Nevertheless, we take s_j/r as an integrated factor and fit its value.

Directly applying numerical techniques to solve the above NLS problem is not feasible in some situations, where the objective function may not be differentiable. As a matter of fact, the shape of the network boundary determines our function of calculating $l_{i,j}$. A non-differentiable network boundary, say, a rectangular field, usually leads to non-differentiable objective function. Traditional numerical techniques like Gauss-Newton method or Levenberg-Marquardt method [15] all require the objective function to be differentiable, thus not applicable in those cases. On the other hand, the direct solution of the NLS problem is not always a stable estimation of the locations of mobile users, due to the measurement errors and model prediction errors. The estimated locations may largely vary between consecutive estimations with different instances of flux observations.

Against such challenges, we propose to approach the mobile user movement with sequential samplings. Under the NLS constraints, we can efficiently filter those outlier samplings and keep a good approximation. With the Sequential Monte Carlo Sampling technique, we are able to cumulate our prior observations on network flux and get constantly refined estimation accuracy.

B. Sequential Monte Carlo Estimation

In our problem, for each mobile user i, there exists a time series of data collections $[t^i_1, t^i_2, ..., t^i_{ki}]$ corresponding to the sequence of its positions $[p^i_1, p^i_2, ..., p^i_{ki}]$. The Sequential Monte Carlo method allows us to represent the real position of the mobile user p^i_j at each instance j with a set of random samples. Those samples are updated iteratively with the importance sampling method. Through the prediction and filtering operations in each round of update, we are able to restrict the samples to the posterior distribution of the mobile user's possible positions.

Let t be the discrete time instances. For mobile user i, it corresponds to the time series of data collections $[t^i_1, t^i_2, ..., t^i_{ki}]$. Let p_t represent the position distribution at time t. We can predict the current position distribution of the mobile user from its previous position, i.e., $P(p_t| p_{t\text{-}1})$. On the other hand, according to our observations on the network flux we get the likelihood of the mobile user's current position with the observation constraints, i.e., $P(p_t| o_t)$. With sequential observations on the network flux evolutions, we iteratively approach the posterior distribution $P(p_t| o_1, o_2, ..., o_t)$. At each stage, we use a set of N random samples P_t to approximate the position distribution p_t. We accordingly update the set of samples as the observed network flux pattern evolves. At each time instance t, P_t is computed with the previous approximation $P_{t\text{-}1}$ and the current observation o_t.

C. Prediction and Filtering

Initially, without any knowledge and constraints on the position of the mobile user we assume a uniform distribution and select the samples uniformly random over the field. At each time step, we predict the possible positions of the mobile sink based on the transition distribution $P(p_t| p_{t\text{-}1})$ and get updated position samples. We then eliminate those predictive samples inconsistent with network flux observations in a filtering phase. In such a process, the sampling distribution gradually approaches the posterior distribution $P(p_t| o_1, o_2, ..., o_t)$.

In the prediction phase, we get the updated set of samples P_t from the previous set $P_{t\text{-}1}$. We assume a weak model to predict the movement of the mobile user, i.e., we do not have any specific information on its mobility pattern (speed, direction, trajectory, and etc.) except the knowledge of its maximum moving speed v_{max}. Thus from any sample position in the previous step $P_{t\text{-}1}(i)$, the possible current position $P_t(i)$ is uniform random within a circular region of radius $v_{max}{\cdot}\Delta t$, where Δt is the time interval between the two consecutive time instances.

```
Algorithm 4.1

Initialization
For each mobile sink
    $t_{last} = 0$
    $P_{tlast} = \{M$ random positions in the field$\}$
    $w_{tlast}(i) = 1/M$
End

Step
For every $\Delta T$ time interval

Observation
    Input = network flux observation vector $F'$
    Record current time $t$

Prediction
    For each mobile sink
        $\Delta t = t - t_{last}$
        $P_t = \{N$ random position samples according to formula 4.2$\}$
    End

Filtering
    Calculate $||F-F'||$ for each of $N^K$ position compositions
    Get top $M$ compositions with least objective values

Asynchronous updating
    For each mobile sink
        If the best fit $s_j/r \to 0$
            Null
        Else
            $t_{last} = t$
            $P_{tlast} = \{M$ position samples in the top $M$ compositions$\}$
            Calculate $\{w_t(i)|(i = 1, 2, \ldots, M)\}$ with formula 4.3
            $w_{tlast} = w_t$
        End
    End

End
```

$$P(p_t \mid p_{t-1}) = \begin{cases} \dfrac{1}{\pi(v_{\max} \cdot \Delta t)^2}, & if \quad d(p_t, p_{t-1}) \le v_{\max} \cdot \Delta t \\ 0, & if \quad d(p_t, p_{t-1}) > v_{\max} \cdot \Delta t \end{cases} \qquad (4.2)$$

After the prediction phase, there are N new samples drawn randomly from the discs centered at previous sample origins, corresponding to increased uncertainty on the movement of the mobile user. Indeed, above mobility model can be further refined if we have more accurate mobility prediction, say, the heading of the mobile user.

In the filtering phase, we eliminate those impossible position samples from P_t to cut down the uncertainty due to the unawareness of mobility. The filtering operation is bound to our network flux observations. For each mobile user i, we estimate the incurred network flux when it is at any of the N possible updated positions. We sum up the flux amounts incurred by all K mobile users and obtain the estimated flux vector F for the n sampling nodes. For all N^K possible combinations of the mobile user positions, we estimate the flux vector F and compare it with the real measurement F'. Since there still exists freedom on the traffic stretches of mobile users, we take s_j/r ($j = 1, 2, \ldots, K$) as integrated factors and fit their values to minimize $||F-F'||$. We are then able to find minimized objective value $||F-F'||$ for each possible combination of the mobile user positions, with specific traffic stretch factor s_j/r. Such observations allow us to filter out those position combinations apart from real measurements by their objective values. We rank the N possible updated positions for each mobile user i, according to their minimum objective values each of which is achieved in N^{K-1} possible combinations. Finally, we keep the top M updated positions for each mobile user and filter out the other possible positions.

D. Importance Sampling

In the prediction and filtering phase of each round, we keep the top M updated positions for each mobile user in P_t and indeed treat them equally in the following round to generate new samples. Such a method may not be the most efficient way to converge our position samples to the posterior distribution $P(p_t| o_1, o_2,\ldots, o_t)$. We slightly alter our sampling method and use importance samplings to achieve faster and more accurate convergence.

Suppose the samples are drawn independently from an importance function. We can measure the importance of each position sample and assign different weights for different samples. We are then able to use these weighted samples to estimate the posterior distribution $P(p_t| o_1, o_2,\ldots, o_t)$. We represent each sample with a duple <$P_t(i)$, $w_t(i)$> which records the position and weight of the sample, and we adopt recursive importance sampling technique to estimate the weight for each updated sample.

$$\begin{cases} w_t(i)' = w_{t-1}(i) \cdot P(o_t \mid P_t(i)) \\ w_t(i) = \dfrac{w_t(i)'}{\sum_{j=1}^{K} w_t(j)'} \end{cases} \qquad (4.3)$$

As formula 4.3 shows, we calculate the weight for each updated sample $w_t(i)$ based on the weight of the original sample $w_{t-1}(i)$ and the posterior probability of the observation at current position $P(o_t|P_t(i))$. We then normalize the weights for all updated samples. With such an iterative calculation, the weighted set <$P_t(i)$, $w_t(i)$> approaches the posterior distribution $P(p_t| o_1, o_2,\ldots, o_t)$. Good approximations for $P(o_t|P_t(i))$ at each round will lead to more accurate estimations on the final posterior distribution and achieve faster convergence. Indeed, the observation o_t at each round comes from the network flux measurements from the n sampling nodes, and the basic sense is that a smaller deviation between the predicted and observed network flux values implies a larger observation probability $P(o_t|P_t(i))$. Thus we use the reciprocal of the minimum objective value $||F-F'||$ of each of the top M sample position to approximate such observation probability.

E. Asynchronous Updating

Recall that in our application context different mobile users collect the updated data from the network at their own wills. For any mobile sink i, there exists a time series of data collections $[t^i_1, t^i_2, ..., t^i_{ki}]$ which is independent with each other. As a matter of fact, the observable updating of their positions is by nature asynchronous. For each round of observing the network flux some mobile users may not happen to collect data from the network and there is a best fit traffic stretch $s_j/r \rightarrow 0$ estimated for each of them in the prediction and filtering phase. In such a case, we will not update the position samples of those mobile users and instead we allow a larger Δt for computing the transition distribution $P(p_t|\ p_{t-1})$ in following rounds. As a result, the samples of different mobile users are asynchronously updated. For each mobile user i, the time interval Δt used to calculate the movement radius $v_{max}\cdot\Delta t$ in formula 4.2 is the time period between two consecutive time points of data collection $t^i_j - t^i_{j-1}$. We show the pseudo code of the Sequential Monte Carlo Estimation in Algorithm 4.1.

V. Evaluations

We do extensive simulations to validate the effectiveness of our approach. We first evaluate the accuracy of locating users inside the network with NLS parameter fitting. We demonstrate the results with various inputs and examine the performance of our approach under different conditions. We then let the internal users move within the network and track their movement by Sequential Monte Carlo Estimation. We vary the number of mobile users and their speed to examine the performance of our approach with different conditions.

Finally, we launch a trace driven experiment with the movement logs of mobile users in Dartmouth Campus data set. We test the efficacy of our approach in tracking those asynchronously updated mobile users.

A. Instant Localization

To demonstrate the basic localization framework that we provide with the fingerprint information of network flux, we simulate a sensor network with 900 nodes on a 30 by 30 rectangular field. The sensor nodes are distributed over the field in perturbed grids [3]. The communication radius for each node is set to be 2.4, resulting in an average degree of 18. We simulate internal users within the field, collecting sensory data from the network. The traffic stretch of each user is randomly selected from 1 to 3. As described in Section 4.1, by doing the NLS fitting on the traffic flux over the network, we are able to approximate the locations of all internal users that are collecting data from the network.

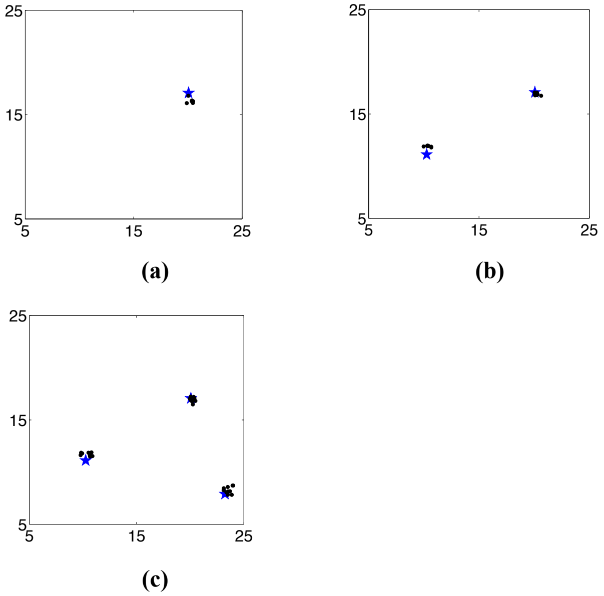
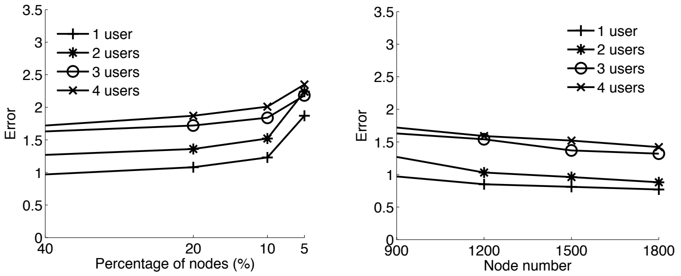


Figure 5. Instant localization cases: (a) one user, (b) two users, and (c) three users.

Figure 5 shows three instant cases in the experiment. In each case, we test 10,000 random location samples for each user and perform NLS fitting to find the top 10 combinations that minimize the objective function $\|F - F'\|$. The true locations of the users are marked with stars. Figure 5 (a) depicts the case where there is only one user inside the network. According to the result, all 10 location predictions concentrate in a small region very close to the true location, with an average error of 0.97, less than 5% of the field diameter. Figure 5 (b) depicts the case where two users reside inside the network. According to the result, the location predictions for both users are still accurate, with an average error of 1.27. Due to the bilateral interference on the traffic pattern of the two users, however, there are some outlier reports that deviate from the true locations. As shown in Figure 5 (b), the largest error is 1.78. Nevertheless, such deviation is rare, and we are able to filter them out by adopting the reports of majority. In Figure 5 (c), there are three users simultaneously collecting data in the network. The traffic pattern over the network becomes more complicated and our approach calculates their locations with an average error of 1.63. The location predictions scatter within a relatively broader area. The largest error reaches up to 2.06. Indeed, as there are more internal users collecting data simultaneously, the network flux introduced by different users cumulate on top of each other, leading to less locating accuracy. Such a problem, however, is largely mitigated in reality when different users collect the updated data from the network at their own wills. In such circumstance, different users collect data asynchronously and at one time interval Δt there are usually quite a small number of active users, letting us easily calculate the locations of them asynchronously. We will later validate this point in our trace driven experiment.

In Figure 6, we evaluate the localization accuracy of our NLS fitting based approach with varied settings. We vary the percentage of sensor nodes that provide us flux samplings, testing the effectiveness of this approach with sparse inputs. For each percentage level, we randomly select the percentage of sensor nodes from the network and use their flux reports to calculate the locations of users.



(a) **(b)**

Figure 6. Localization accuracy (a) against percentage of sampling nodes and (b) against network density.

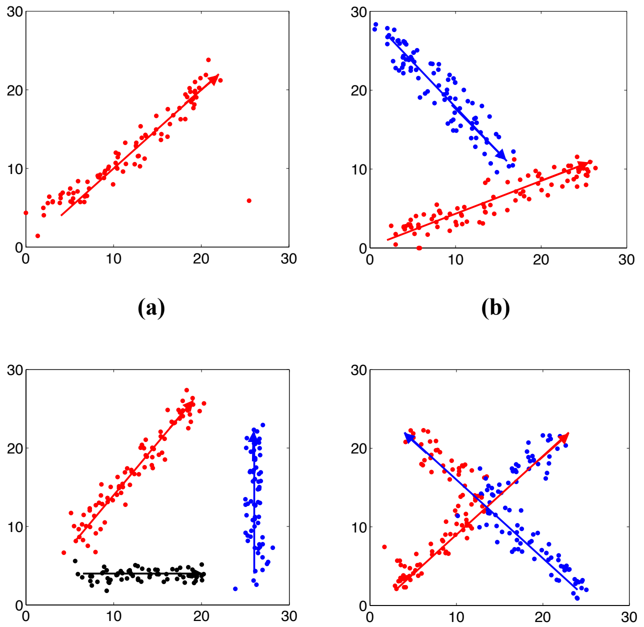


Figure 7. Instant tracking cases: (a) one user, (b) two users, (c) three users, and (d) two users(crossing)

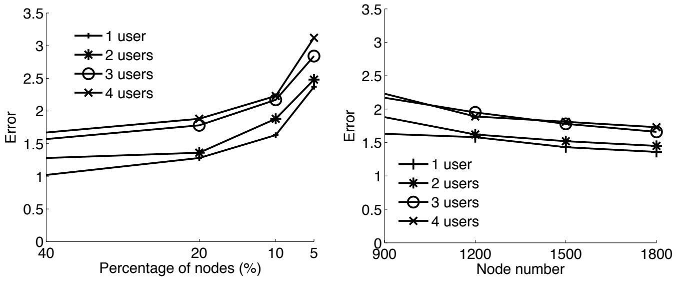
In Figure 6 (a), we show how the localization accuracy varies with the percentage of sensor nodes we use. Consistent with our intuition, as the percentage of sampling nodes drops, the localization error increases. Nevertheless, the results prove that our approach is robust with sparse inputs. The localization error keeps low even when we only use the reports from 10% nodes. A dramatic increase of error happens when we further lower the usage of node samplings to below 5%. The number of simultaneous internal users affects the localization error. When we employ 10% nodes, our approach achieves localization error of 1.23 for one user, 1.52 for two users, 1.84 for three users and 2.01 for four users. We then vary the number of nodes deployed in the field from 900 to 1800, resulting in different network densities. For this set of simulation, the node reports we use is fixed at 90. As Figure 6 (b) depicts, the localization error decreases as the network density rises. That is probably because in a denser network the proposed network flux model approximates the real network traffic more accurately, as we previously discussed in Section 3. The impact of network density, however, is fairly limited. The localization error does not significantly change with the network density.

B. Tracking Mobile Users

In this simulation, we let mobile users move within the field and track their moving trajectories by our Sequential Monte Carlo Estimation based approach. The basic settings are the same as previous ones. In the Monte Carlo sampling process, we select N=1000 random samples every time and keep the top M=10 samples as the updated representatives for the location of each user. At this stage we assume that all mobile users simultaneously collect data with the same time interval, so we are able to test how accurate our approach will work with the complex traffic pattern assembled with multiple users. The maximum moving speed of each user is restricted below 5 per detection interval Δt, resulting in a resampling area of radius 5 each round.

Figure 7 presents us several instant cases where different number of mobile users move along different trajectories. Our algorithm continuously calculates the locations of each user in 10 rounds. We point out the 10 location representatives for each user each round. The arrow lines indicate the movement trajectories of the mobile users. Figure 7 (a) presents the simplest case where there is only one mobile user moving within the field. We can easily find that the location estimations converge to the real trajectory from initial deviations as more and more flux inputs are gathered. Finally, the average error is limited below 2. Figure 7 (b) depicts the situation where there are two mobile users moving simultaneously. Our approach still captures their moving trajectories with high accuracy and the location estimations approach the real trajectories as time evolves. In Figure 7 (c) we depict the case of three mobile users. The tracking accuracy decreases a bit due to the complexity of the network flux introduced by multiple users but still maintains high. A particular interesting case is depicted in Figure 7 (d), where the trajectories of two mobile users intersect and the two users come across with each other in their movement. We find that when two mobile users meet, our algorithm with solely network flux input can only detect the locations of them but cannot distinguish their identities, i.e., our algorithm might mix up their location samplings, introducing errors for the successive predictions. As we observe from this figure, the blue and red dots mix up at the intersection point and then follow the trajectories of the other user. Nevertheless, our algorithm calculates the accurate locations for the two users although their identities are mixed. Thus our algorithm preserves the basic trajectories of the mobile users.

We test the accuracy of our approach with different percentage of flux samplings and against different network densities. We measure the error of the location estimation of each user in the final round and depict the results in Figure 8. As shown in Figure 8 (a), the tracking accuracy does not vary



(a) (b)

Figure 8. Tracking accuracy (a) against percentage of sampling nodes and (b) against network density.

much until the percentage of sampling nodes drops below 5%, which is consistent with what we observe in the localization scenario. Although a large number of flux samplings help to provide high accuracy, using only 10% nodes still provides us acceptable accuracy. In Figure 8 (b), we depict how the tracking error varies with the network densities. The number of nodes deployed in the field is varied from 900 to 1800 and the node reports we use is fixed at 90. Similar with the situation in the localization scenario, the network density does not significantly affect the tracking accuracy, although a denser network provides more accurate approximation with the network flux model.

C. Trace Driven Experiment

To examine the performance of our approach in practice, we launch a trace driven experiment with the Dartmouth Campus data set. We use the mobility data set v1.3 extracted form the "syslog" portion of the Dartmouth Wireless-Network Traces [8]. The data include traces from April 2001 to June 2004 recording the wireless APs associated with different network interface cards at different time. There are around 500 APs distributed within the Dartmouth Campus and we use the 50 of them in a rectangular region as landmark references for the locations of mobile users. Figure 9 shows a part of those APs distributed in the campus. The mobility data set records a sequence of APs each user uses at different time, so by concatenating the locations of those APs we are able to figure out a mobility path for each user. The mobility path, however, reflects the movement behaviors of each user in a relatively long period, e.g., one data instance records the AP association of one network interface card for more than 6200 hours. To make compact movement trajectories we intercept a segment from each record and compress the timeline by a factor of 100. We divide the test field into a 30 by 30 grid and simulate 900 sensor nodes deployed with in the field.

We deploy sensor nodes in two ways, into perturbed grids and purely random. While the former deployment represents more regular conditions, the latter one stands for more variability. We do our experiment 10 runs, each time with 20 randomly chosen mobility records representing 20 mobile users. According to the locations and timelines recorded in the mobility traces, mobile users asynchronously collect data from the sensor network. We run our asynchronous updating algorithm (algorithm 4.1) to calculate their instant locations repeatedly. We measure the tracking error between the calculated locations of each mobile user and its movement trajectory and summarize the average error in Figure 10. In Figure 10 (a), we vary the percentage of reporting nodes and observe the tracking accuracy. Consistent with our previous simulation results, with perturbed grid deployment the tracking error is quite limited below 3 when we use more than 10% node reports, i.e., the error is less than 5% of the diameter of the test field. The tracking error smoothly increases when we use less node reports. The tracking error with purely random deployment, however, is about 1.5 times that with perturbed grids. The deviation of errors is also enlarged in such a more variable setting. In Figure 10 (b), we vary the maximum speed of each mobile user. The direct result is that in the prediction of our algorithm the disc area for resampling is enlarged and the sample locations drawn are scattered more widely. The results in Figure 10 (b), however, show that our approach is robust to

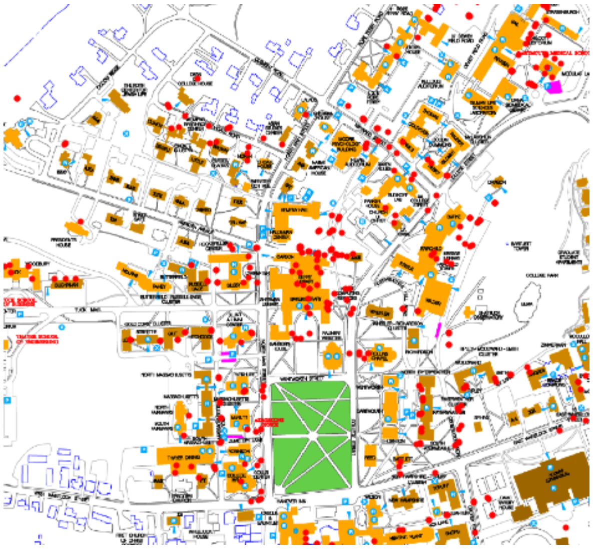
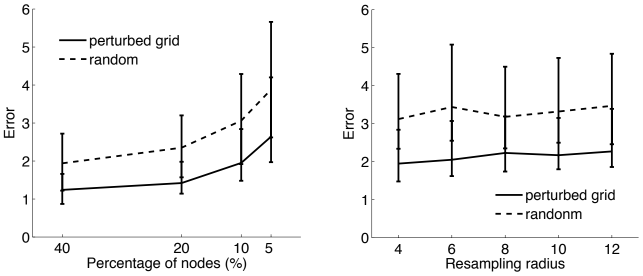


Figure 9. APs distributed in Dartmouth Campus.



(a) (b)

Figure 10. Tracking accuracy in the trace driven experiment (a) against percentage of sampling nodes and (b) against the maximum speed of mobile users.

such increased uncertainty. The tracking error for both perturbed grid deployment and random deployment keeps relatively stable with a slight increase as the maximum moving speed increases.

An interesting observation is that, although in each run of the experiment there are 20 mobile users coexisting within the field the performance of our algorithm does not degrade much. That is benefiting from the asynchronous operations of the mobile users. Due to the asynchronous updates, for most of the small time instances, there are only a small number of mobile users issuing data collections, leading to the ease of calculating the traffic flux of limited complexity.

VI. Conclusions and Future Work

In this paper, we demonstrate that the mobile users within a sensor network take the risk of leaking their location privacy. The network flux provides us fingerprint information about the mobile users inside the network. We propose a flux model that approximates the network flux within the network. Using the NLS fitting algorithm we are able to analyze the network flux and gradually calculate the locations of mobile users with Sequential Monte Carlo Estimation. Through passively sniffing at a small set of nodes in the network any adversary can easily locate the mobile users and track their movement. Indeed, our study reveals the potential threat in protecting the location privacy of mobile users from malicious entities. Future work includes exploring effective countermeasures against such a threat, e.g., reshaping the network traffics to prevent malicious detection.

Acknowledgement

This research was supported in part by China NSFC Grants 60933011, the National Basic Research Program of China (973 Program) under Grant No. 2006CB303000, the National Science and Technology Major Project of China under Grant No. 2009ZX03006-001, and the Science and Technology Planning Project of Guangdong Province, China under Grant No. 2009A080207002.

References

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," In Proceedings of IEEE INFOCOM, 2000.

[2] A. R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, vol. 2, 2003.

[3] J. Bruck, J. Gao, and A. A. Jiang, "MAP: Medial Axis Based Geometric Routing in Sensor Network," In Proceedings of ACM MobiCom, 2005.

[4] S. Garfinkel, A. Juels, and R. Pappu, "RFID Privacy: An Overview of Problems and Proposed Solutions," *IEEE Security and Privacy*, vol. 3, 2005.

[5] D. Goldenberg, P. bihler, M. Gao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang, "Localization in Sparse Networks using Sweeps," In Proceedings of ACM MobiCom, 2006.

[6] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computer*, vol. 34, pp. 57 - 66, 2001.

[7] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks with Multidimensional Scaling," In Proceedings of IEEE INFOCOM, 2004.

[8] D. Kotz, T. Henderson, and I. Abyzov, "Trace set dartmouth/campus/movement (v1.3)," 2005.

[9] B. Kusy, A. Ledeczi, and X. Koutsoukos, "Tracking Mobile Nodes using RF Doppler Shits," In Proceedings of ACM SenSys, 2007.

[10] B. Kusy, H. Lee, M. Wicke, N. Milosavljevic, and L. Guibas, "Perdictive QoS Routing to Mobile Sinks in Wireless Sensor Networks," In Proceedings of IEEE/ACM IPSN, 2009.

[11] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor Network Navigation without Locations," In Proceedings of IEEE INFOCOM, 2009.

[12] H. Lim and J. C. Hou, "Localization for Anisotropic Sensor Networks," In Proceedings of IEEE INFOCOM, 2005.

[13] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. J. Guibas, "Composable Information Gradients in Wireless Sensor Networks," In Proceedings of IEEE/ACM IPSN, 2008.

[14] S. Madden, M. J. Franklin, and J. M. Hellerstein, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks," In Proceedings of OSDI, 2002.

[15] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for Nonlinear Least Squares Problems," Tech. rep., 2004.

[16] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," In Proceedings of ACM SenSys, 2004.

[17] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *ACM Wireless Networks*, vol. 10, pp. 701-710, November 2004.

[18] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AOA," In Proceedings of IEEE INFOCOM, 2003.

[19] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 User Fringerprinting," In Proceedings of ACM MobiCom, 2007.

[20] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," In Proceedings of ACM MobiCom, 2000.

[21] A. Savvides, C. Han, and M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," In Proceedings of ACM MobiCom, 2001.

[22] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, and K. Frampton, "Sensor Network-Based Countersniper System," In Proceedings of ACM SenSys, 2004.

[23] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking Moving Devices with the Cricket Location System," In Proceedings of ACM MobiSys, 2004.

[24] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, 1992.

[25] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*: Springer Verlag, 1997.

[26] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet," In Proceedings of ACM SenSys, 2004.