Plausible Mobility Inference from Wireless Contacts UsingOptimization

Alirea k.Monfared Mostafa Ammar Ellen Zegura

***Abstract*—Recording locations of mobile nodes and connectivity between them is essential in the study of various types of wireless networks, including but not limited to opportunistic networks. Working with mobility information gives an extra degree of freedom in the study of such networks, while in a contact log, a lot of information like the locations of the nodes and their velocities is irretrievably lost. On the other hand, contact information can be measured more easily compared to detailed locations of the nodes. In this paper, we present a solution for inferring mobility from contact information. This technique uses a simple non-linear optimization approach to suppress the problem as a feasibility problem with fundamental constraints on the mobility of the nodes. In this paper, we introduce our technique and its underlying assumptions, and discuss applicability of more than one input to enhance the performance of inferred mobility.**

***Index Terms*—mobility trace, contact trace, plausible mobility**

I. Introduction

STUDIES of mobile wireless protocols benefit from real-world traces that measure and record node locations over time. These *mobility traces* can be combined with radio models to produce network connectivity for simulation studies. Measuring node locations with fine-grained time and space granularity is challenging, however. Outdoor positioning systems are sensitive to weather and obstacles, while indoor positioning systems generally rely on availability of WiFi or other signal infrastructure that may not be available.

As an alternative, *contact traces* measure and record pair-wise node connectivity over time and do not require accurate positioning systems. Because they come from radio transmissions, contact traces are able to capture environmental effects on communication capability, such as obstacle effects, that are missing from mobility traces. Contact traces can record on-off timing information about links, hence they can be more compact than mobility traces that must record data at every time instant.

Contacts are typically recorded by using Bluetooth or WLAN in an infra-structureless mode. All other devices in range are seen as a contact [1]. Collecting contact traces is challenging, however, as it requires instrumenting nodes with collection capability and conducting a well-planned and coordinated collection “run”.

Alireza K.Monfared is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, e-mail: alireza@gatech.edu.

Mostafa Ammar is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA, e-mail:ammar@gatech.edu.

Ellen Zegura is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA, e-mail: ewz@cc.gatech.edu.

While recording contact opportunities, a lot of information is irretrievably lost. No information on exact location an velocity of the nodes can be retrieved from a contact trace. As time elapses, the amount of this information decreases and it becomes more and more difficult to even relatively locate the nodes. Such traces are only representative of the particular context in which they are measured. For example, a WiFi radio contact trace seems not to be appropriate to study a scenario possessing the same characteristics in every aspect, except a Bluetooth radio without running the entire trace collection experiment again which carries a significant organizational overhead in preparing, distributing and collecting contact loggers, not to mention non-technical issues related to legality and participant consent. In a similar fashion, every small change in the contact scenario necessitates repetition of the whole collection process.

Contact trace collection and analysis have been reported in the literature and have been used extensively to study mobile networks for various settings ranging from a campus scenario [2][3] to a conference [4] to urban scenarios [5] or a disaster area. Aschenbruck et al.[6] has summarized these types of datasets.

We are interested in the problem of inferring a plausible mobility trace from a contact trace. By “plausible”, we mean a trace that is consistent with the contacts, and not originating from a given radio model. In general, there may be multiple mobility traces that are consistent with a given trace. We do not attempt to enumerate them. Although this plausible mobility might not match the original locations, it matches to a mobility trace that could have generated the same contact trace, or a trace very similar to it. It can be discussed that the exact locations of the nodes is not required in most the experiments that deal with studying mobility or modifying contacts , unless the researcher has access to a “valid” mobility trace presenting the same characteristics.

Coming from the contact domain to the mobility domain, may provide researchers with opportunities not present with the contacts alone, such as transforming the settings of the original data collection, one that assumes a different radio, for example.

Our proposed algorithm is based on optimizing a mobility trace using the limited information from the contacts. It operates by solving a feasibility problem for a small set of fundamental constraints on mobility of the nodes. We will describe the details of the algorithm in this report and we have made the code publicly available [CITATION NEEDED]

Contributions of this work are summarized as follows:

- An algorithm to infer mobility information based on

contact traces that enables richer understanding of mobile networks.
- Application of multiple inputs for the proposed algorithm to enhance re-usability of the output trace.

The rest of this paper is organized as follows [TO BE COMPLETED]

II. Background

In this section, we define the basic concepts of mobility and contacts as they have been used in our proposed algorithm, and provide the background knowledge required for understanding the context and technical details of our mobility inversion algorithm. We start this section by a few definitions.

Contact Trace: A contact trace is a log of contact-start and contact-end events, each labeled with a time stamp and with the identifiers of the two nodes involved in the contact event. Mathematically, contact traces can be represented using a sequence of graphs showing the connections over time, which are called *evolving graphs* [7]. We are interested in contact traces collected in the course of an experiment. In such experiments, mobile devices equipped with a wireless interface are attached to moving entities (e.g., people or vehicles). These mobile devices sample their environments periodically and log the presence of neighboring nodes. At the conclusion of the experiment these logs are collected and a contact trace is constructed from them.

Contact-start events occur when one of the devices receives information from the other and contact-end events occur when transmission stops. Many aspects of the radio environment can affect these events. In general, a contact trace becomes meaningful when studied along with a *radio model.* A radio model specifies the circumstances under which two nodes are defined to be in contact. In the simplest model, two nodes are defined to be in contact whenever their distance is less than a given transmission range, and disconnected otherwise. This circular transmission range model is considered very simplistic due to many reasons, including the existence of obstacles, lack of symmetry in real contacts, inequality of signal strength inside the assumed transmission circle, effects like fading, shadowing, etc. [8], but this model has been used extensively due its simplicity and the fact that it allows to focus on other, perhaps more significant factors in the experiment.

Mobility Trace*:* A mobility trace is a log of locations of nodes over time. Similar to a contact trace, a mobility trace can also be asynchronized or synchronized, synthetically modeled or originating from real data collected using GPS equipped devices or association to the base stations of a 3G/WiFi network.

The mobility inversion problem combines a radio model, and a contact trace to obtain a consistent mobility trace showing their locations using no further information[1].

Few solutions have been proposed for this specific problem. Wang et al.[9] solves the same problem by dividing it into smaller sub-problems, and relying on decreasing number of possibilities. Our analysis of such approach has proven that the complexity of such solution grows exponentially with increasing number of nodes, and hence is only applicable to small scenarios. Ristanovic et al.[10] suggests a solution that relies on knowledge of some anchor node with fixed locations which can give a rough estimate of location of mobile Bluetooth devices that have a low range. The immediate downside is the dependence on the location of devices and the underlying technologies used.

Whitbeck et al. [11] has proposed a complete algorithm for the mobility inversion problem inspired by the work in dynamic graph drawing. They define three fundamental constraints on the mobility of the nodes as follows

$$||x_i(t+\Delta t) - x_i(t)||_2^2 \leq v_{max}\Delta t$$

$$r - 2v_{max}min\{t - p_{ij}^{\uparrow}, n_{ij}^{\downarrow} - t\} \leq d_{ij}(t) \leq r$$

$$r \leq d_{ij}(t) \leq r + 2v_{max}min\{t - p_{ij}^{\downarrow}, n_{ik}^{\uparrow} - t\}$$

Where $x_i(t)$ and $d_{ij}(t)$ are the location of node i and distance between nodes i and j at time t respectively, $p_{ij}^{\uparrow}$and $n_{ij}^{\uparrow}$are the previous and the next time that node i and j will be in contact , $p_{ij}^{\downarrow}$ and $n_{ij}^{\downarrow}$are the previous and the next time that node i and j will be lose their contact, all relative to the current time t, and r denotes the transmission range of the nodes. Note that these definitions rely on the assumption of circular transmission range model.

The first constraint states that no node can move faster than a maximum speed v_{max} and the other two confine the distance between two nodes based on their previous and future contact opportunities. The authors aslo define, an attractive force, similar to the classical "spring" force on each node from all of its contacts, as well as a repulsive force, similar to the classical "Coulomb" force, as well as a "drag" force to prevent nodes going excessively away from each other when borders are not obvious. The equilibrium of these forces on each node determines its location. Together, this gives a relative set of locations satisfying the given contacts. A major disadvantage of this approach is the necessity of a smart choice of many constants involved in the calculation of the forces. Since there is no rule of thumb for many of these parameters, the performance of the application becomes dependent of the specific scenario under study, and it may become a series of trial and errors when such information is not available.

III. Mobility Inversion Algorithm

In this section, we define the problem and its solution in a mathematical framework. Let N be the number of nodes that reside in a field of size $L \times L$. We assume each node has a location in d dimensions, with $d = 2$ in our simulations. A contact trace in this setting, corresponds to an $N \times N \times T_m$ matrix of adjacencies of its corresponding evolving graph [7] where T_m is the number of time steps . A mobility trace is is a $d \times N \times T_m$ matrix, where its element $(:, i, t)$[2] is the location

[1]These concepts will be more formally defined in the next section

[2]Throughout this document, we use the notation of colons,$(\ldots, :, \ldots)$, to represent all possible numbers that can come in a specific vector location

of node i at time t in both dimensions. Table I describes the parameters used in the above description.

Parameter Name	Description
N	Number of nodes
R_i	Transmission Range of i^{th} Input
v_m	Maximum Speed
T_m	Trace Length in Time
d	Dimensions of the problem (2 in our case)
L	Size of the Simulation Field (square) Side

Table I: Summary of basic parameters used in the algorithm

Our algorithm does not use any information on initial locations or assumption about the mobility model of the nodes. The only assumptions made are the following:

- Two nodes x_i and x_j are in contact once their distance, $||x_i - x_j||_2^2$ is less than a given radio range, R, and are out of contact if this distance is larger than R.
- Nodes move slower than a given maximum speed v_m, i.e. , the distance between the location of a node i at time t and $t+1$, $||x_i^t - x_i^{t+1}||_2^2$ is smaller than $v_m \times \Delta t$ where Δt is the difference between snapshots.
- Nodes are always confined inside the mobility field.

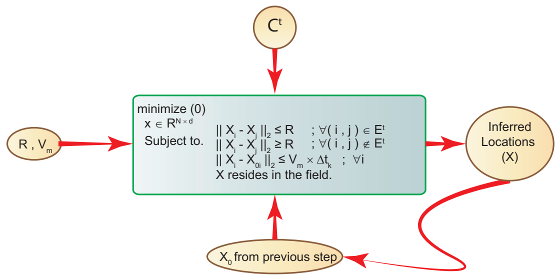
We will describe the technical details of the algorithm and its variants in the next section.

A. Single Input Mobility Inversion

In this section, we describe the algorithm for a single input contact trace, and then we generalize this to an arbitrary number of inputs in section III-B. This front end is depicted in figure 1, where we use the notation $G = (V, E)$ for the evolving graph corresponding to the input contact, which is a time-series of graphs $G^t = (V^t, E^t)$. We use the notation $G \Leftrightarrow C$ to show that the evolving graph G has the corresponding adjacency matrix C , which is a contact trace as defined above, this means that the graphs G^t have adjacency matrices $C(:, :, t)$. The algorithm works in an iterative manner to find the locations of the nodes at each time step of the contact trace.

Algorithm 1 Basic mobility inversion algorithm with one input contact trace

```
Input: Contact Trace $C_{N \times N \times T_m}$, Transmission range $R$, Maximum speed $v_m$, number of dimensions $d = 2$
$X_{final} \leftarrow \mathbf{0}_{d \times N \times T_m}$
Find a set of initial locations $X_0$.
$X_{final}(:, :, 0) \leftarrow X_0$
for $i$ :=1 to $T_m$ do
    solve the nonlinear optimization problem in figure 1 with $G^t \Longleftrightarrow C(:, :, t)$ , $X_0$, $R$, $v_m$
    $X_{final}(:, :, t) \leftarrow X$
    $X_0 \leftarrow X$
end for
Output: $X_{final}$
```



C^t
R , V_m
minimize (0)
$x \in R^{N \times d}$
Subject to.
$\| X_i - X_j \|_2 \le R \quad ; \forall(i,j) \in E^t$
$\| X_i - X_j \|_2 \ge R \quad ; \forall(i,j) \notin E^t$
$\| X_i - X_{0i} \|_2 \le V_m \times \Delta t_k \quad ; \forall i$
X resides in the field.
Inferred Locations (X)
X_0 from previous step

Figure 1: Optimization algorithm takes a contact traces along with its corresponding radio range and a value of maximum speed to infer a mobility in an iterative manner. Inferred locations for each time step are given as an initial point of the solver for the next step.

To complete our description of Algorithm1, we need to clarify how we find the initial value of X_0 and how we solve the nonlinear optimization of figure 1.

To solve the optimization problem, proposed in figure 1, which is done in every iteration of Algorithm 1, we use Limited-memory Broyden–Fletcher–Goldfarb–Shanno (l-BFGS-B) [12]. The l-BFGS-B is a variant of the Broyden–Fletcher–Goldfarb–Shann (BFGS), an approximation of Newton's method that seeks the stationary point of a function by finding the zero of its derivative. The advantage of the BFGS to its alternative methods is that it does not need to evaluate the Hessian matrix of the second derivatives. Instead, the Hessian is approximated using rank-one updates of gradient evaluations. The basic operation of the BFGS is described in Algorithm 2.

Algorithm 2 Basic BFGS algorithm.

```
Input: Objective function $f(X)$, approximate initial Hessian $B_0$(can be identity matrix), maximum number of iterations $max-iter$
for $k$ :=1 to $max-iter$ do
    Solve for direction,$p_k$ : $B_k p_k = \Delta f(X_k)$
    Perform line search with acceptable step size $\alpha_k$ to perform the update: $X_{k+1} = X_k + \alpha_k p_k$
    Set : $s_k := X_k + \alpha_k p_k$
    $Y_{k+1} = \Delta f(X_{k+1}) - \Delta f(X_k)$
    $B_{k+1} = B_k + \frac{Y_k Y_k^T}{Y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$
end for
Output: $Y_k$
```

The l-BFGS variant is particularly suited to problems with very large numbers of variables and BFGS-B variant handles simple box constraints. Also not that in the algorithm, l-BFGS-B needs an initial point to seek this stationary point, which is the value X_0 given to it in algorithm 1.

To solve the equation for the first iteration, we use the same method as above, except that only constraints corresponding to ranges, and residence of nodes inside the simulation field will be considered and the maximum speed constraint, which necessitates a reasonable set of "previous" locations, will be ignored. Since l-BFGS-B still needs an initial point to seek the stationary point of the function, we construct a distance matrix by finding the number of hops among every pair of nodes in $G^0 \iff C(:,:,0)$, and multiplying it by the transmission range R, i.e. we assume two nodes m hops from each other are at a distance $m \times R$, and apply the classical MDS [13] to yield a decent initial point which can be then passed to the version of the optimization problem that ignores the speed constraint to find the first value of X_0, which is initial location of nodes as well, i.e. $X_{final}(:,:,0)$.

B. *Multiple Input Mobility Inversion*

In this section, we extract the mobility inversion problem, in case that there are multiple inputs. This can be useful to enhance the quality of the inferred mobility when more information about the contacts is available. Assume that we have contact traces $C_1, C_2, ..., C_n$. Each contact trace C_i is a $N \times N \times T_m$matrix as describe in section III. Also assume their corresponding transmission ranges to be $R_1 \leq R_2 \leq ... \leq R_n$. The algorithm presented in 1, can be easily generalized to accept multiple inputs. The revised version operates as follows:

Algorithm 3 Basic mobility inversion algorithm with one input contact trace

```
Input: Contact Trace $\{C_1, C_2, ..., C_n\}_{N \times N \times T_m}$ Transmission ranges $\{R_1 \leq R_2 \leq ... \leq R_n\}$, Maximum speed $v_m$, number of dimensions $d = 2$
Preprocess contact traces$\{C_1, C_2, ..., C_n\}$
$X_{final} \leftarrow \mathbf{0}_{d \times N \times T_m}$
Find a set of initial locations $X_0$.
$X_{final}(:,:,0) \leftarrow X_0$
for $i$ :=1 to $T_m$ do
    solve the nonlinear optimization problem in figure 1 with $\{G_1^t, G_2^t, ..., G_n^t\} \iff \{C_1(:,:,t), C_2(:,:,t), ..., C_n(:,:,t)\}$ , $X_0$, $R$, $v_m$
    $X_{final}(:,:,t) \leftarrow X$
    $X_0 \leftarrow X$
end for
Output: $X_{final}$
```

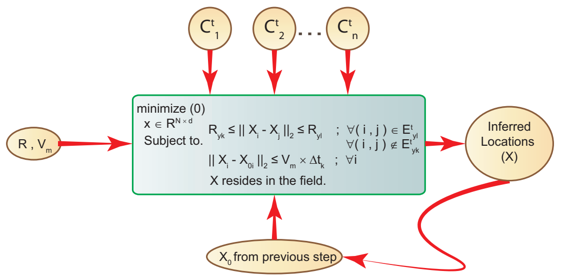

C^t_1
C^t_2
C^t_n
R , V_m
minimize (0)
$x \in R^{N \times d}$
Subject to.
$R_{yk} \leq \| X_i - X_j \|_2 \leq R_{yl}$; $\forall(i,j) \in E^t_{yl}$
$\forall(i,j) \notin E^t_{yk}$
$\| X_i - X_{0i} \|_2 \leq V_m \times \Delta t_k$; $\forall i$
X resides in the field.
Inferred Locations (X)
X_0 from previous step

Figure 2: Optimization algorithm takes multiple contact traces along with their corresponding radio ranges and a value of maximum speed to infer a mobility in an iterative manner. Inferred locations for each time step are given as an initial point of the solver for the next step.

Pre-processing is needed to make sure that the contact traces $C_1, C_2, ..., C_n$ are compatible. This step handles the situations in which two nodes are connected in a contact trace with a smaller transmission range at some time step but the same nodes are not connected in another contact trace with a larger range. Note that the necessity of this pre-processing is tied with our assumption of circular transmission ranges. A more complex radio model might not see this step necessary. In our algorithm, to pre-process, we first inspect the contact traces starting from C_1 and then C_2 and so on to C_n. At any points, if two nodes are not connected in a contact trace C_i with transmission R_i at a certain time, we need to force all contact traces C_j with transmission ranges $R_j \geq R_i$ not to have a connection between the same nodes at the same time[3]. A follow-up inspection is then performed, starting from C_n, and then $C_{n-1}$to C_1. At any points, if two nodes are connected in a contact trace C_i with transmission R_i at a certain time, we need to force all contact traces C_j with transmission ranges $R_j \leq R_i$ to have a connection between the same nodes at the same time[4]. The pre-processing ensures that there is no contradiction in the contact traces. Note that such contradictions are possible, if we use real traces, and skip the pre-processing due to environmental effects, etc.

The optimization problem to be solved in the iterations of algorithm 3 is the second revision to the basic algorithm. In this case, for every two nodes, we can limit their distance at every time t to two values. One is the transmission range R_k corresponding to the contact trace $C_k$with the largest transmission range that still shows the two nodes not to be connected, i.e. $C_k(i,j,t) = 0$, and the other is the transmission range R_l corresponding to the contact trace $C_l$with the smallest transmission range that still shows the two nodes to be connected, i.e. $C_k(i,j,t) = 1$. Note that in general case, we may have $R_k = 0$, i.e. the two nodes are connected in all contact traces for that specific time instance, and we can also have $R_l \to \infty$, i.e. that the two nodes are never connected in any contact trace for that specific time instance.

Except the above mentioned revisions, the optimization problem is solved with similar techniques as the ones discussed in section III-A.

C. *Discussion*

In this section, we briefly review the main advantages of our mobility inversion algorithm, compared to the force based algorithm of Whitbeck et al.[11]. We will also point out the drawbacks of the algorithm. We will discuss the performance metrics in the next section.

Simplicity of the front-end is the first advantage of our algorithm. The force-based algorithm not only relies on the input contact trace, but also many parameters that need to be set. These include the rigidity constant K and the damping factor τ in the attractive force, the intensity constant G , cutoff distance d_{max},and parameters α and ϵ_0 that affect the intensity

[3]If those nodes are labeled l and k, respectively, then if $C_i(l,k,t) = 0$, we force $C_j(l,k,t) \leftarrow 0$,for all $R_j \geq R_i$

[4]If those nodes are labeled l and k, respectively, then if $C_i(l,k,t) = 0$, we force $C_j(l,k,t) \leftarrow 1$,for all $R_j \leq R_i$

of the repulsive force with distance, as well as the strength parameter D of the drag force. There is no general rule for setting these parameters, and for the algorithm to perform well, these need to be set in a trial and error fashion. On the other hand, the optimization algorithm does not need any external parameters other than a maximum speed and transmission range, which are natural to consider for every scenario and are present in the force-based algorithm nevertheless.

Optimization algorithm can take one or more contact traces, their corresponding radio ranges, and an upper bound on speed of movements as input and infer a mobility trace that can generate these contacts. This has not been foreseen in the force-based algorithm of Whitbeck[11], but can be possibly incorporated with some modifications.

A drawback, present in both algorithms is the radio model. It has been discussed in the literature, many other factors including but not limited to presence of obstacles, fading, path-loss, etc affect the radio model [8]. Despite this, the optimization mobility inversion algorithm is designed in a modular manner so that any radio model can be integrated into it provided that the an well-behave penalty, possibly in the form of a non-linear inequality, corresponding the connectivity status of the nodes is suggested[5]. Since the solver at the heart of our algorithm, BFGS, can solve any nonlinear optimization problem, a new contact model that can be described mathematically, and can be derived from a contact trace, possibly with additional information, can replace the circular model, without affecting the other modules.

A second future improvement to the algorithm concerns complexity. The BFGS solver used in the algorithm does not scale linearly with the number of nodes, and this can be a serious drawback to the performance of the algorithm on large traces. The authors do not know of any solver or feasible modification at the moment that can address this completely.

IV. Performance Evaluation

In this section, we evaluate the performance of the mobility inversion algorithm proposed.

A. Metrics

Before presenting the evaluations, we need to define a performance metric to measure how “good” is the performance of the algorithm on a trace. To do this, in the very general framework ,we need to derive measures of comparison from the inputs and the outputs of the algorithm. Such comparison can be done in three levels as follows.

The lowest level of evaluation is done with mobility-mobility comparison. Assuming that we have access to the mobility trace from which the input contact trace originated, which is only viable for test purposes, we can compare the original input mobility trace and the output mobility trace for each location, and report a mean squared error over time:

$$E_{location} = \frac{1}{N^2 \times T_m} \sum_{i,j,t} ||X_{original}(i,j,t) - X_{inferred}(i,j,t)||^2$$

[5]In the circular model, for example, this penalty is that the distance between two connected nodes is smaller than a transmission range, and the distance between two disconnected nodes is larger than this range.

We do not expect a good matching in mobility nevertheless, as the proposed feasibility does not have a unique solution, and we have shown that it actually has many solutions which makes it almost impossible to pick the desired output mobility among all these possible solutions.

In the next level, we compare the contact traces. While the input to our algorithm is a contact trace, we can infer an output contact trace with the proper transmission range that the algorithm has used at the output and a proper radio model We can then compare the two corresponding evolving graphs over time based on the number of missing links, i.e. present in the input contact trace and missing in the inferred contact trace, extra links, i.e. only seen in the inferred contact trace, and the sum of these two numbers. While a low number of link errors in this level is very desired to be seen and is sufficient to have a successful mobility inference, it is not necessary for our purposes. These errors can be defined as a function of time as follows:

$$E_{missing\ link}(t) = \frac{\sum_{i,j}(C_R(i,j,t) \wedge (\neg C'_R(i,j,t)))}{\sum_{i,j}(C_R(i,j,t) \wedge \mathbf{1})} \times 100$$

$$E_{extra\ link}(t) = \frac{\sum_{i,j}((\neg C_R(i,j,t)) \wedge (C'_R(i,j,t))}{\sum_{i,j}(C_R(i,j,t) \wedge \mathbf{1})} \times 100$$

$$E_{total\ link}(t) = \frac{\sum_{i,j}(C_R(i,j,t) \oplus C'_R(i,j,t))}{\sum_{i,j}(C_R(i,j,t) \wedge \mathbf{1})} \times 100$$

Where C_R and C'_R are contact traces corresponding evolving graph of ground truth, aka the input to the algorithm, and derived contact which comes from the inferred mobility, and $\mathbf{1}$ is a matrix of all ones of the size $N \times N \times T_m$.$\neg$, $\oplus$ and $\wedge$ represent element-wise logical NOT, XOR and AND respectively. Note that for such comparison to be reasonable, we need both contact traces to correspond to the same transmission range R.

In the third level of comparisons, we use the definitions from Chen et al.[14] and Bui et al.[7] to compare journeys, i.e. space-time paths between node pairs over time. In its simplest form, we can categorize node pairs at each time instant to three groups:

- S-pairs: Two nodes that are always reachable in space at every instant of time after the current time.
- ST-pairs: Two nodes that are reachable within a journey of limited duration after the current time. [6]
- N-pairs: two that are not reachable after the current time. [7]

We can then compare the percentage of S-,SP-, and N-pairs at every instant of time for the input and inferred contact trace. This shows the connectivity characteristics of the trace, and can be used as a metric to compare the two traces.

[6]As discussed by Chen et al. [14], A journey, is a space-time path in an evolving graph that connects two nodes over the course of time. It’s represented as $J = (R, R_\delta)$ in an evolving graph G, where $R = e_1, e_2, ..., e_k$, the sequence of edges it traverses, and $R_\delta = \delta_1, \delta_2, ..., \delta_k$ the corresponding time instants of edge traversal. R_δ must be such that each edge traversed is in the evolving graph at the time of traversal.

[7]Note that S-pair,ST-pair, and N-pairs are a function of time, e.g., that two nodes can be an ST-pair at a time t_1, but an N-pair at another time t_2

B. Results

In our experiments, we use three mobility models:

- Random Waypoint (RWP)[15], which is often used for simulation purposes in mobility literature despite its known limitations [**?**]. In this model speed, direction, and destination of each mobile node is chosen randomly and independently of other nodes.
- Reference Point Group Mobility (RPGM) [16] which organizes mobile hosts into groups according to their logical relationships with each group having a logical center known as reference center whose movement is followed by all nodes in the group.
- Disaster Area Model (DA) [6] which is based on an analysis of tactical issues of civil protection and has added features like heterogeneous area-based movement, obstacles, and joining/leaving of nodes.

The experiments setting consists of 50 nodes and a duration of 1000 time steps. RWP and RPGM span over a $1000 \times 1000 m^2$ field, with a maximum speed of $10m/s$, while DA spans over a $350 \times 350\, m^2$ field, with a maximum speed of $20m/s$,.

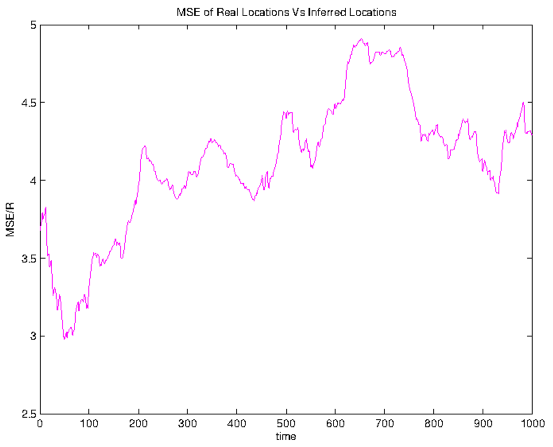


Figure 3: Error in location inference for RWP over time. Vertical axis shows the MSE of such error devided by the radio range to make the numbers globally comparable.

Figures 3,4, and 5 show the MSE of the errors in location prediction ($E_{location}$is section IV-A). We do not expect the algorithm to find the exact locations of the nodes, since no initial information has been used, but as it is obvious the errors in the case of RPGM and DA are increasing over the time. This does not happen with RWP. We believe that this behavior is due to erratic behavior of RWP. In RPGM and DA, the number of possibilities that match the inferred location decrease over the time, and there is a higher correlation between the current and previous locations. This is much more random in case of RWP. For this reason, in the former, our algorithm inevitably has to get closer to the original locations over the time, and pick one of the fewer possible location sets that are compatible with the input contacts.

Figures 6, 7, and 8 show the errors resulted in comparing the input contact trace, and the contact trace resulted from the inferred mobility.As we can see these errors have some spikes in case of RPGM and is more smooth for RWP and DA. This

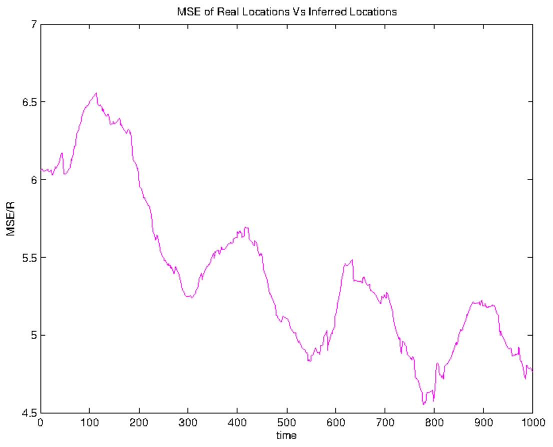


Figure 4: Error in location inference for RPGM over time. Vertical axis shows the MSE of such error devided by the radio range to make the numbers globally comparable.

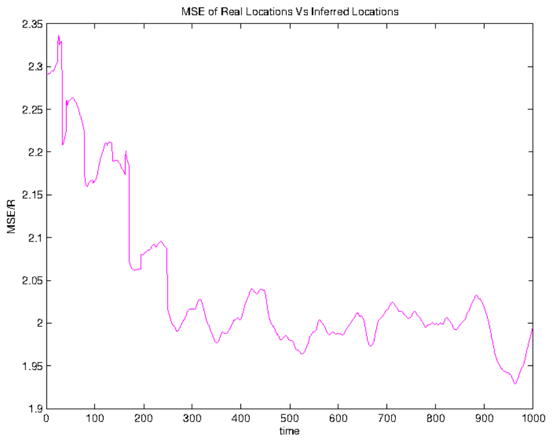


Figure 5: Error in location inference for DA over time. Vertical axis shows the MSE of such error devided by the radio range to make the numbers globally comparable.

is mostly due to the grouped nature of RPGM and presence of obvious communities in that. Also it's obvious that the errors for DA are much smaller than the other tow models. This is due to the fact that most nodes in DA model are wandering in a dense area, and number of connections are surprisingly high compared to the RWP and RPGM, resulting in a denser connectivity graph.

Figure 9, compares the input contact trace and the trace inferred from the output mobility with the same radio range for RWP, in the sense that at each moment of time, the number of nodepairs classified as S-pair, ST-pair, and N-pair are counted and plotted in a box plot. Figure 9a shows this count for the input contact trace, and figure 9b shows it for the inferred trace. This last results shows that despite errors present in the contact level comparison, when compared for connectivity characteristics, our algorithm finds an output mobility trace that has the same properties with a high precision. This ensures that the inferred set of locations can be used in any experiment, instead of the original location that had resulted in the input contact trace, and yet the same behavior will be observed.

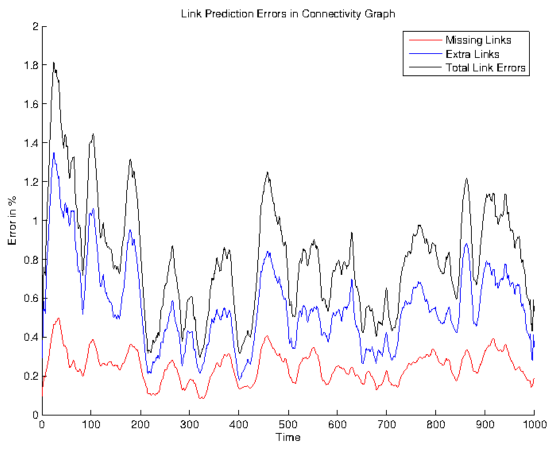


Figure 6: Error in contact inference for RWP over time. Extra links are missing in the input, but present in the contacts inferred from the output mobility. Missing links are present in the input, but missing in the contacts inferred from the output mobility. Total link errors is the sum of these.

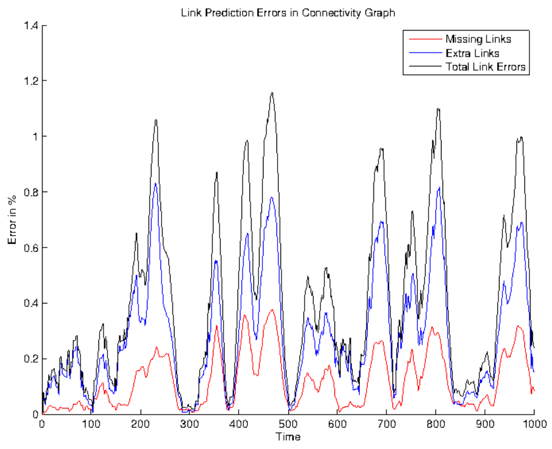


Figure 7: Error in contact inference for RPGM over time. Extra links are missing in the input, but present in the contacts inferred from the output mobility. Missing links are present in the input, but missing in the contacts inferred from the output mobility. Total link errors is the sum of these.

V. Conclusion

We have presented an algorithm to infer mobility from contacts. Our algorithm accepts a contact trace that indicated the connections among nodes over time. Without any extra information about the locations of these nodes, we infer a set of locations, aka a mobility trace, that could have generated these contacts. Through performance evaluations, we have shown that this inferred mobility shows comparable connectivity characteristics compared to the original locations from which the input contact trace has come. This opens up the opportunity to use these locations in any experiment, and avoid the expensive process of collecting location information. It also gives rise to an ongoing research to use these locations to infer contact traces for other scenarios, which results in transformation of a contact trace.

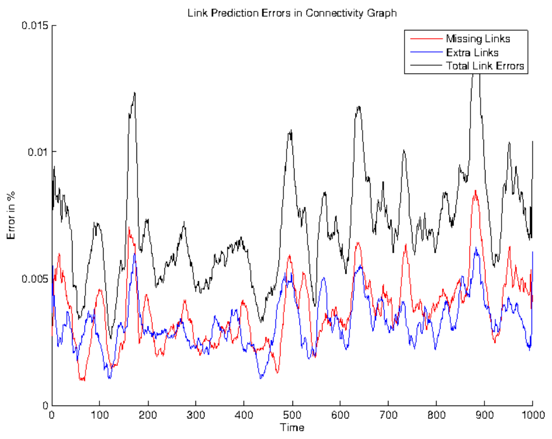


Figure 8: Error in contact inference for DA over time. Extra links are missing in the input, but present in the contacts inferred from the output mobility. Missing links are present in the input, but missing in the contacts inferred from the output mobility. Total link errors is the sum of these.

Acknowledgment

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1117190.

References

[1] N. Aschenbruck, A. Munjal, and T. Camp, “Trace-based mobility modeling for multi-hop wireless networks,” *Computer Communications*, vol. 34, no. 6, pp. 704–714, 2011.

[2] P. Meroni, S. Gaito, E. Pagani, and G. P. Rossi, “CRAWDAD data set unimi/pmtr (v. 2008-12-01),” Downloaded from http://crawdad.cs.dartmouth.edu/unimi/pmtr, Dec. 2008.

[3] V. Srinivasan, M. Motani, and W. T. Ooi, “CRAWDAD data set nus/contact (v. 2006-08-01),” Downloaded from http://crawdad.cs.dartmouth.edu/nus/contact, Aug. 2006.

[4] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29),” Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006, May 2009.

[5] A. Natarajan, M. Motani, and V. Srinivasan, “Understanding urban interactions from bluetooth phone contact traces,” *Passive and Active Network Measurement*, pp. 115–124, 2007.

[6] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, “Modeling mobility in disaster area scenarios,” *Performance Evaluation*, vol. 66, no. 12, pp. 773–790, 2009.

[7] B. Bui Xuan, A. Ferreira, A. Jarry *et al.*, “Evolving graphs and least cost journeys in dynamic networks,” 2003.

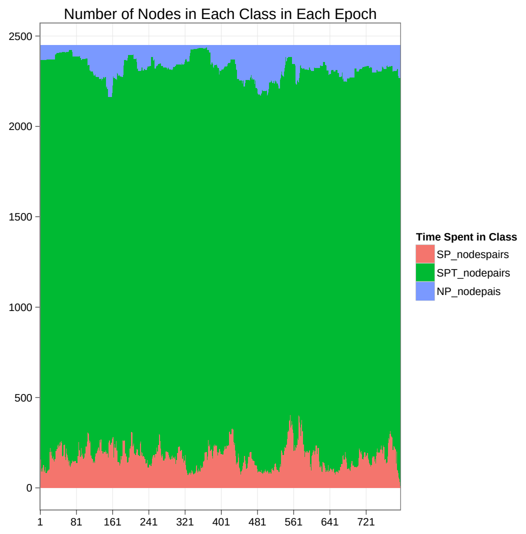
[8] D. Kotz, C. Newport, and C. Elliot, “The mistaken axioms of wireless network research’, Darmouth College,” *Computer Science Department*, 2003.

[9] P. Wang, Z. Gao, X. Xu, Y. Zhou, H. Zhu, and K. Zhu, “Automatic inference of movements from contact histories,” in *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*. ACM, 2011, pp. 386–387.

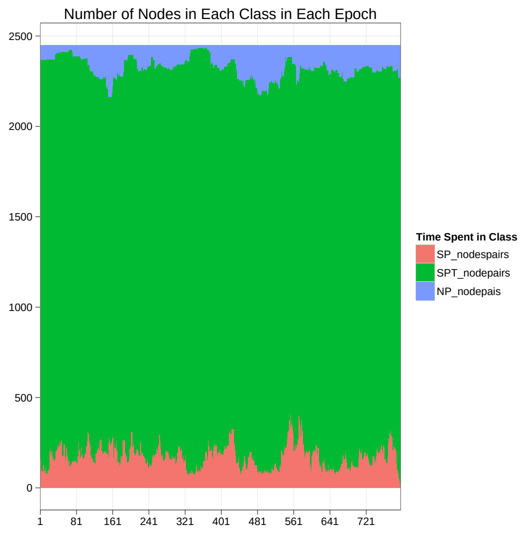
[10] N. Ristanovic, D. Tran, and J. Le Boudec, “Tracking of mobile devices through Bluetooth contacts,” in *Proceedings of the ACM CoNEXT Student Workshop*. ACM, 2010, p. 4.

[11] J. Whitbeck, M. De Amorim, V. Conan, M. Ammar, and E. Zegura, “From encounters to plausible mobility,” *Pervasive and Mobile Computing*, vol. 7, no. 2, pp. 206–222, 2011.

[12] C. Zhu, R. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.



(a) Input contact trace



(b) Contact trace inferred from output mobility

Figure 9: Comparison of nodepair classifiaction for RWP over time.

[13] G. Colliat, "Olap, relational, and multidimensional database systems," *ACM Sigmod Record*, vol. 25, no. 3, pp. 64–69, 1996.

[14] Y. Chen, V. Borrel, M. Ammar, and E. Zegura, "A framework for characterizing the wireless and mobile network continuum," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 5–13, 2011.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[16] R. Roy, "Reference Point Group Mobility," *Handbook of Mobile Ad Hoc Networks for Mobility Models*, pp. 637–670, 2011.

[17] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," 1980.

[18] Y. Shang and W. Ruml, "Improved MDS-Based Localization." in *INFOCOM*, 2004. [Online]. Available: http://dblp.uni-trier.de/db/conf/infocom/infocom2004.html#ShangR04

[19] Y. Koren, "On Spectral Graph Drawing." in *COCOON*, ser. Lecture Notes in Computer Science, T. Warnow and B. Zhu, Eds., vol. 2697. Springer, 2003, pp. 496–508. [Online]. Available: http://dblp.uni-trier.de/db/conf/cocoon/cocoon2003.html#Koren03

[20] M. Broxton, J. Lifton, and J. A. Paradiso, "Localization on the pushpin computing sensor network using spectral graph drawing and mesh relaxation." *Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 1–12, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/sigmobile/sigmobile10.html#BroxtonLP06

[21] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 46–54.

[22] P. Biswas, T.-C. Liang, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization." *TOSN*, vol. 2, no. 2, pp. 188–220, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/tosn/tosn2.html#BiswasLWY06

Appendix

In this appendix, we survey other techniques to infer the locations of wireless nodes, given various amount of information.

A closely related problem to our mobility inversion, localization has been a well-studied problem in various disciplines of networking. In wireless sensor networks, the problem also called the "graph realization" problem, is generally stated as follows:

Given a graph $G = (V, E)$ such that V is the set of n vertices (nodes), and D is an $n \times n$ matrix of pairwise distances between nodes, find a $d - dimensional$ embedding that preserves D.

Note that in our problem, we deal with an evolving graph $G = \{G^t(V^t, E^t) \quad \forall t\}$, which is a series of graphs whose information are correlated. Also, we do not assume access to the distance matrix, D, in our algorithm.

Next, we will briefly review three widely used proposed solutions for graph realization problem, which is proved to be NP-Hard in the general case[17].

Multidimensional Scaling (MDS) is a statistical technique used for exploring the similarities or dissimilarities in data. To find an embedding $X = [x_1 ... x_m]^T$[8]of n vertices in $\mathbb{R}$, and given a known distance matrix D, multidimensional scaling, seeks to minimize a stress function as follows:

$$\min_{x_1 \dots x_m} \sum_{i<j} \quad (||x_i - x_j|| - d_{ij})^2$$

Classical MDS requires distances between every pair of nodes. Shang et al. [18] have proposed to use shortest paths computed from a connectivity matrix instead of the distance matrix in their MDS-MAP[9]. MDS yields a relative map of locations of the nodes, and given d anchor nodes (nodes with known locations), this can be turned into an absolute map with the target embedding, X. MDS has the disadvantage of pulling nodes towards the center of the area, and thus performs poorly on irregularly-shaped graphs. To overcome this problem, Shang et al. [18], have produced a version of the algorithm, MDS-MAP(p) in which MDS is applied for the subgraph of a given number of hops, p, around each node to produce a local map for that node, and then these local maps are merged to produce a global relative map.

[8] $[]^T$ represents transpose operation on a vector.

[9]Connectivity matrix is defined similar to an adjacency matrix. For a graph $G = (V, E)$ the $n \times n$ connectivity matrix A is defined such that $A_{ij}$is 1 whenever nodes i and j are within transmission range of each other and 0 otherwise.

Spectral Graph Drawing (SGD) is another solution proposed for the graph realization problem. SGD also defines a minimization problem based on the Laplacian matrix, L. This matrix is defined as follows:

$$L = K - W$$

$$W = [w_{ij}] = \frac{1}{1 + d_{ij}} \quad or \quad e^{-d_{ij}}$$

$$K = diag(k_{ii}) = \sum_{j \epsilon N(i)} w_{ij}$$

W is a weighting matrix, inversely proportional to the pairwise distances D, and K is the diagonal matrix of node degrees. SGD solves the following optimization problem to find X:

$$X^* = \underset{X}{argmin}[X^T L X]$$

$$s.t. \quad X^T X = 1$$

$$X^T \mathbf{1}_n = 0$$

The first constraint in the above equation ensures that the trivial solution of all nodes in the same place is not chosen in favor of the others. The selection of variance, $X^T X$, to be equal to 1 in this constraint is arbitrary. The second constraint, ensures the invariance of the solution under transitions by setting a mean of zero. Koren [19] argues that like classical MDS, SGD tends to draw the nodes to the center and hence has a poor performance on irregularly-shaped graphs. A variant of SGD, called degree normalized spectral graph drawing (DN-SGD), updates the second constraint of original SGD with weights proportional to degrees, i.e. $X^T K X = 1$. SGD and DN-SGD have been shown to be solvable by finding the generalized eigenvalues of the matrices L and K. Broxten et al. [20] have shown that for DN-SGD to perform reasonably on static sensor network localization, a two-step localization, followed by a post-processing to filter outliers is required.

Finally, the localization problem has been formulated as a Semidefinite program (SDP), a specific type of convex optimization which can be solved by interior methods. An SDP feasibility is as follows:

$$find\ X \in \mathbb{R}^{(n+2)\times(n+2)}$$

$$\begin{aligned} s.t. &(1;0;\mathbf{0})^T X (1;0;\mathbf{0}) = 1 \\ &(0;1;\mathbf{0})^T X (0;1;\mathbf{0}) = 1 \\ &(1;1;\mathbf{0})^T X (1;1;\mathbf{0}) = 2 \\ &(0;e_i - e_j)^T X (0;e_i - e_j) = d_{ji}^2 \ \forall (j,i) \in V_x \\ &(a_k - e_j)^T X (a_k - e_j) = d_{jk}^2 \ \forall (j,k) \in V_a \\ &X \succcurlyeq 0 \end{aligned}$$

$V_x \subseteq V$ is the set of nodes with unknown locations, and $V_a \subseteq V$ is the set of nodes with known locations (anchors). and e_i is a vector of all zero entries except i^{th}-entry, and $a_k \quad \forall k = 1...m$ is the locations of the anchor nodes. The last constraint requires X to be a positive semidefinite matrix[10].Biswas et al. [21], [22] have discussed details of using semidefinite programming for solving the localization problem.

[10]A matrix, M, is positive semidefinite, if $z^T M z \geq 0$ for all non-zero column vectors z