

Model-based Tracking for Mobile Ad Hoc Networks

Xuming Lu, Murat Demirbas

Computer Science and Engineering Department, SUNY at Buffalo, Buffalo, NY 14260

xuminglu@gmail.com, demirbas@buffalo.edu

Received: July 30, 2011 Accepted: August 21, 2011 DOI: 10.5296/npa.v3i2.830

Abstract

We consider the problem of tracking of a mobile target node in a mobile ad hoc network (MANET) set-up. We find a Gradient model alone is usually not very efficient, whereas a precise Markov model which includes transition probabilities is too hard to achieve. We propose a generic tracking framework for online tracking applications, by integrating a Gradient model of the target's proximity and an online statistically estimated Markov model of the target's likely direction. We show PMBT achieves a short catching path with a high success rate. PMBT is a probabilistic online tracking algorithm that computes information utilities at each step, and then chooses the next step toward the target based on the maximum expected utility. Our algorithm avoids the need to maintain a tracking data structure (such as a hierarchical directory look-up structure) and the need to send periodic update messages about the target's location. Simulation results show, by taking a hybrid approach that integrates a gradient model and a Markov model, our algorithm significantly outperforms both gradient-based and Markov approaches alone.

Keywords: MANET, PMBT, Gradient Model, HMM, target.

1 Introduction

Mobile ad hoc networks (MANETs) [1, 2] do not require a fixed infrastructure support for communication and routing, and hence find many applications in urban warfare, disaster recovery, and large-scale sensing/surveillance scenarios. These scenarios become more feasible with the progress in mobile robotics platforms [3, 4] and cheap ultraportable wireless devices. In particular, a noteworthy progress recently is the kick off of the One Laptop Per Child (OLPC) program. OLPC program is currently shipping more than 600K laptops to developing countries for education. These laptops use IEEE 802.11 based ad hoc wireless communication to self-organize themselves into a mesh network. Tracking of a mobile target is an important service for MANETs that enables routing of a message to the mobile target [1, 5, 6]. Such target tracking in MANETs is challenging because the tracking protocols need to deal with---in addition to the mobility of the target--- the mobility of the intermediate nodes that maintain a track toward the target. Since the intermediate nodes that maintain the tracks to the target and relay protocol messages are also allowed to be mobile, maintaining a distributed location lookup/tracking directory over the network is hard, let alone doing it in a bandwidth-efficient and distance-sensitive manner.

Existing work on target tracking in MANETs mainly lie in two categories: structure-based approaches and structure-free approaches. The first type of protocols dynamically maintain certain structures, such as trees or overlay graphs, such that a message may simply follow the structure to reach the target [7, 8, 9, 10]. Although most of these protocols provide deterministic location services, they have the following drawbacks: 1) Maintaining a structure is costly in terms of communication/energy is not scalable in MANETs setting, and 2) The target's location needs to be periodically updated in the structure, which introduces a large communication overhead.

In structure-free protocols, nodes take local decisions by making prediction over available knowledge to generate a path on demand for a tracking message to be forwarded to the target. Gradient based and/or prediction based approaches are classic examples of this type [11, 12, 13]. Structure-free approaches use much less communication, gradient can be generated effortlessly due to the mobility of the nodes [14]. However, structure-free approaches are based on predictions rather than location look-up, and are prone to tracking failures (dropping of the target message) or taking long delivery paths to the target.

Our goal in this paper is to improve on the effectiveness of structure-free tracking approaches. To this end, we introduce a hybrid Probabilistic Model Based Tracking (PMBT) framework that integrates a Gradient model of the target's proximity and an online statistic Markov model of the target's likely direction. The goal of this protocol is to reach closer to the same effectiveness level of the structure-based frameworks without the periodic updating costs of the target. Our main contributions are as follows:

- We introduce the PMBT protocol that, for the first time, integrates a Gradient model and Markov model into a unified model for tracking in MANETs. PMBT is a

probabilistic online tracking algorithm that computes information utilities at each step, and then chooses the next step toward the target based on the maximum expected utility. By comparing PMBT's performance with that of the Gradient model and the Markov model under different circumstances, we show that PMBT outperforms both.

- We prove the convergence conditions of PMBT protocol and further provide the A-PMBT protocol capable of tuning the parameters adaptively by utilizing the confidence of the gradient. A-PMBT leverages the weight of the Gradient model and Markov model adaptively, and yields better performance than PMBT.

- We provide a lightweight implementation abstraction for all these models. Our abstraction is based on a loose synchronization the state of nodes in each cell and a polite gossip mechanism for forwarding the tracking messages.

The rest of the paper is organized as follows: Section 2 introduces the Markov model, the Gradient model and the PMBT model, and then compares their pros and cons through analysis. Section 3 describes the lightweight abstraction we use for the distributed implementation of these frameworks. We present the simulation results in Section 4, and review the related work in Section 5. Finally, we conclude the paper in Section 6.

2 Probabilistic Model Based Tracking

After clarifying the assumptions in our set up, we first present a summary of the specialized Hidden Markov Model based tracking and the Gradient based tracking protocols. We then present our protocol PMBT, which is a hybrid of these two approaches.

2.1 Assumptions

We assume a dense mobile ad hoc network. Nodes reside on a two dimensional plane, with their localized coordinates (x,y) available. The field is divided into grid regions, overlaying a logical virtual cell layer over the network. We set the cell size such that all nodes between neighboring cells are within one hop distance, as in the setup in [15].

Nodes do not have a global view of the entire network. In contrast to the set up at [16] that assumes that nodes can monitor the target continuously in the field, we do not make any such assumptions. Nodes can sense/detect the target in a binary manner only when the target is in the same grid cell as the node. Historical data are cached by the mobile nodes who have encountered tracking targets. To save energy, they are not exchanged between nodes.

The computation of the optimal track in Hidden Markov Model (HMM) is very costly especially when the number of nodes becomes large. The solution is to simplify the prediction process given that the movement of target can only take place between four neighboring cells. What's more, utility information is constructed in a distributed manner using a weighted average of the gradient and the transition probability. The gradient results from node mobility: a node encountering the target saves the target's location and sets the

gradient to be one that descends as it gets older. In the following, we first introduce the HMM model and the Gradient model, followed by the formalization of PMBT model.

2.2 Hidden Markov Model

HMM has been used for solving the 2D plane target tracking problem in [17, 18]. A discrete HMM can be applied when the division of the region is finite. The approach we describe here discretizes the target region into cells and formalizes the transition matrix in an online probabilistic model rather than deterministic transition parameters.

The HMM model for tracking contains the following parameters:

1. the possible states $S = S_1, S_2, \dots, S_N$, each cell corresponds to one state.
2. the transition probabilities $A = a_{ij}(1 \leq i \leq N)$, where a_{ij} is defined as the

transition probability of moving from state S_i to S_j , that is $a_{ij} = P(q_k = S_j | q_{k-1} = S_i)$.

Geometrically, a_{ij} is only meaningful when the states S_i and S_j are neighboring cells. To simplify, we consider only four transitions, moving up, down, left and right (It is straightforward to allow more transitions such as moving upleft, upright, downleft and downright). The rest of the elements in matrix A are all zeros. Figure 1 shows the discretized target space and the constrained HMM transition model.

3. an initial distribution of the target in each state $\Pi = \Pi_i(1 \leq i \leq N)$.

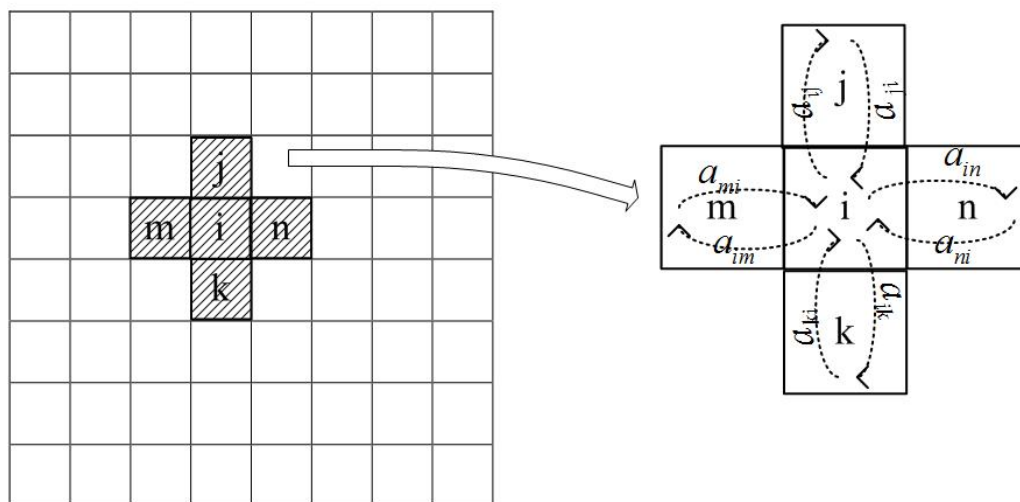


Figure 1: Target space discretization and constraints in HMM transition model.

Viterbi algorithm can be used to find the most likely track (state sequence) given the

observations O_1, O_2, \dots, O_n (i.e., signals). Sensor arrays are used to get these data: each node reports an observation at every discrete time. Viterbi algorithm finds the most probable path at each step by maximizing likelihood, yet it is extremely computational costly and time consuming for a large number of cells (states). However, given the fact that the transition probability a_{ij} is only limited to adjacent cells, each cell can work independently in each step. The entire path can be achieved by simply joining the subtrack at each cell. In our paper, we use this distributed implementation, the details of which are discussed in Section 3.

Although the HMM model has been used in target tracking applications, it adds some constraints when applied to distributed mobile ad hoc networks. Firstly, the algorithm relies on the target's prior probability distribution, which is hard to acquire precisely; secondly, the HMM model assumes each state $P(s(t_n)) = P(s(t_n)|s(t_{n-1})) \cdot P(s(t_{n-1}))$, predicting the next move based only on the last location, which may not be correct in some cases; lastly, although in most real world cases, a target's movement is quite limited (i.e., limited on roads), for randomized mobility models (i.e., random waypoint mobility model), HMM cannot provide good prediction as it is inherently unpredictable.

The transition information in each cell are loosely synchronized to reduce coordination overhead. A node enters a new cell, it drops old information and picks up the information in the cell by sending a DATA REQUEST message. We discuss the lightweight implementation in more detail in Section 3.

2.3 Gradient-based Model

The gradient-based model we discuss here shares similar concepts with [14], and differs in gradient construction and message forwarding. The intuition behind this model is that the history of target events provides gradient indication toward the target. This gradient is maintained not by communication among nodes, but solely by the node mobility inherent in the MANET.

When a node detects the target, its gradient value is set to 1 and the location and timestamp also remembered by the node. This gradient information tells us the location of target sometime ago. We use an exponential decreasing function to keep decreasing this value at that node with the passing of time. This is inspired by the decay law in physical phenomena that an event's effect decreases exponentially to the distance of the event, i.e., $N(d) \propto \frac{1}{e^d}$. More specifically, we construct the gradient with respect to time t as equation 1:

$$\mathbf{g}(t) \propto \begin{cases} \mathbf{1} & t = 0 \\ e^{-t} & 0 < t < T \\ \mathbf{0} & t \geq T \end{cases} \quad (1)$$

where T is the lifetime of an event: when the record is refreshed (a node detects the target), $\mathbf{g}(t)$ becomes 1; and the record will be dropped after T to avoid redundancy. This model yields a gradient distribution where the nearby nodes to the target tend to have higher gradient values whereas farther away nodes hold lower gradient values(or zero). This is

because in most cases, the distance a node travels is proportional to time t . For instance, in Brownian motion, the distance traveled is proportional to the square root of the time t . It is worthwhile to mention that our gradient generation process relies on the mobility of sensor nodes and does not involve any message exchanges between nodes.

Figure 2 shows the gradient distribution under two different mobility models - the random walk model and the random waypoint model, where all the nodes are mobile. We can see that the distribution simulates the decay law well enough in general for gradient based tracking algorithms to perform effectively, however some local maximas are also observed. Due to the uneven mobility of the nodes, the information distribution is not smooth. Local maxima, local minima, or flat regions where information gradients are either the same or zero, may exist. In practice, other factors such as obstacles or node failures, also causes such discontinuity in the decay distribution. To overcome the local maxima problem, the gradient based model selects the best move from the neighbors, i.e., from i to j , even if the gradient $g(j, t) < g(i, t)$. The tracking message marks the local maxima node to prevent the message being looped back by neighboring nodes. This way tracking messages can escape from local maxima and eventually be forwarded to the target.

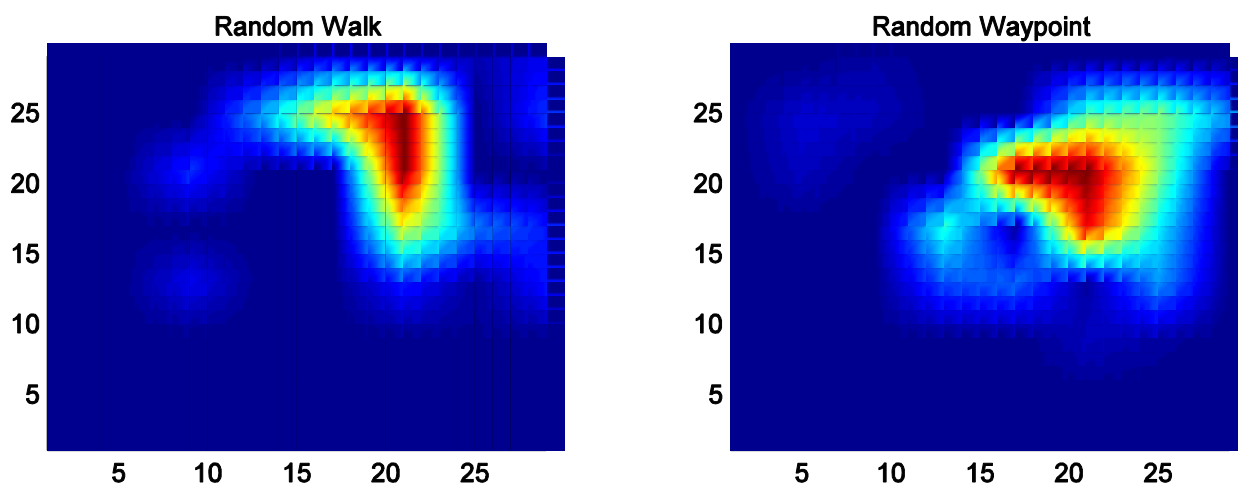


Figure 2: Information decay space distribution with mobile speed 2m/s. Both the random walk model and the random waypoint model show similar decay behavior.

In [9] we show that a cell abstraction can be achieved easily given the assumption that the coordinates of any nodes are available (a node does not need any message exchange with other nodes to decide which cell it belongs to). Gradients are based on nodes rather than cells, and they are not exchanged between nodes. Hence we need a mechanism to decide which node in the cell is responsible to forward tracking messages. To this end, we use a polite gossip message forwarding scheme that does not require querying of all the nodes in a cell to decide which node should be used for forwarding the tracking message further. For this scheme, we associate the gradient information with an application layer backoff timer, such that nodes with higher gradient values have better chances to respond and relay the tracking message. We discuss this novel design in more detail in section 3.

Note that this is a greedy based searching algorithm, therefore cannot guarantee an optimal path due to the lack of global knowledge. After each step, the query message sets closer to the target with high probability.

2.4 PMBT Model (Probabilistic Model Based Tracking Model)

So far we have discussed two tracking models: the gradient-based model and the HMM model. The gradient-based model provides a progressive tracking scheme, but is quite limited to the freedom and the percentage of mobile nodes: in an extreme case where all nodes are static, the target's information cannot be distributed, hence the model will not work well. The limitation of the HMM model, as we have mentioned before, is that it has to be trained before a prediction and is not suitable for random movement targets.

HMM only works when the target is highly predictable, and gradient model does not work well when the nodes are static as our gradient distribution relies on the node mobility. In order to reap the advantages of both the gradient-based model and the HMM model while minimizing their disadvantages, we use information utility for measuring the contribution of nodes in each cell. As both the transition probability and gradient measures the likely direction of the target, our information utility for cell \mathbf{i} is defined as equation 2:

$$\varphi(\mathbf{i}) = \alpha\phi_1(\mathbf{a}_{ij})\cos(\theta_{ij}) + \beta\phi_2(\mathbf{g}(\mathbf{i}, \mathbf{t})) \quad (2)$$

Here, $\phi_1(\mathbf{a}_{ij})$ is the normalized weight of the Markov transition probability where \mathbf{a}_{ij} is the best move for HMM model, and $\phi_2(\mathbf{g}(\mathbf{i}, \mathbf{t}))$ is the weight for the gradient at cell \mathbf{i} (as shown in Figure 1). We illustrate the definitions for θ in Figure 3. In the figure, θ_{ij} is the angle between the line from cell \mathbf{i} to the estimated target \mathbf{t} and the line from cell \mathbf{i} to \mathbf{j} if we decide to take a forward from cell \mathbf{i} to cell \mathbf{j} . Note the exact θ is not known since the nodes may not know the current location of the target. Our θ is based on the gradient information certain period ago when that gradient is initially created. The factor $\cos(\theta_{ij})$ indicates the projection of $\phi_1(\mathbf{a}_{ij})$ on the gradient $\phi_2(\mathbf{g}(\mathbf{i}, \mathbf{t}))$ as they are not in the same direction. As $-1 \leq \cos(\theta) \leq 1$, $\phi_1(\mathbf{a}_{ij})$ has positive impact on the gradient when $0 \leq \theta \leq \pi/2$ and negative impact on the gradient when $\pi/2 < \theta < \pi$.

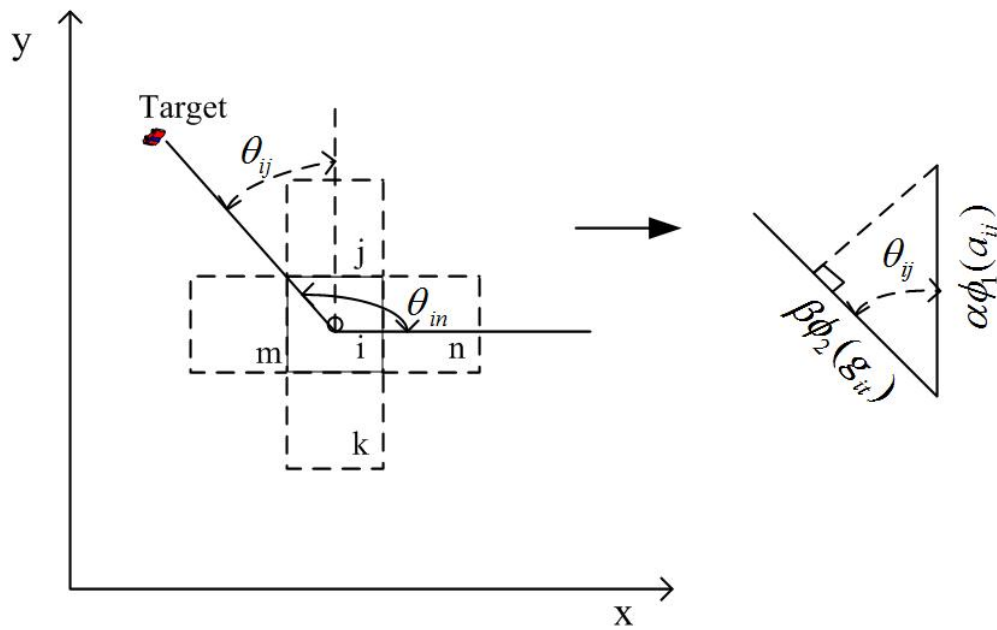


Figure 3: The figure shows the construction of the information utility. The coefficient $\cos(\theta_{ij})$ indicates the projection of $\phi_1=(a_{ij})$ on the gradient $\phi_2(g(i, t))$, α and β are the weights for each model.

Given the information utility, the message forwarding strategy is as follows: when a message arrives at a cell, the nodes in the cell make a decision based on the calculated information utility as to which cell the message should be forwarded. Nodes in a cell hold same a_{ij} but different g_{it} . Hence nodes in the same cell have various utility values at each snapshot. The node with highest utility decides the next forwarding direction. Our implementation (Section 3) avoids communication between nodes in this decision process by associating the utility value to an application layer backoff timer.

Theorem 1 (Convergence Theorem). *PMBT is guaranteed to deliver all tracking messages to the target's location in a connected network given that β is not zero and satisfies the following condition: \forall step from i to j , $\phi_2(g(j, t)) \geq \phi_2(g(i, t)) + \frac{\alpha}{\beta} \phi_1(1) \cos(\theta) (0 \leq \theta \leq \pi)$.*

Before we formally prove the convergence theorem, let's first take a look at the degenerate case where β is zero. According to our discussion, this is the case that only considers the Markov model. It is possible that the target takes unusual actions, i.e., takes a move from cell i to cell k , although statistically $a_{ij} > a_{ik}$. This implies that purely relying on Markov information does not guarantee that all tracking messages can reach the target.

Proof. The proof of the theorem can be divided into two parts: Case 1, $\alpha = 0$, hence the model becomes gradient model. Since each step i to j , $g(i, t) < g(j, t)$, $g(i, t)$ is a monotonically increasing function from the source to the target, so does $\phi_2(g(i, t))$. By the

definition of $g(t), \forall (0 \leq t \leq T), 0 \leq g(t) \leq 1$, and only at target position $g(t) = 1$. This guarantees the tracking messages will only stop at the target; Case 2, $\alpha \neq 0 (0 < \alpha < 1)$.

Since \forall move from i to j , $\phi_2(g(j, t)) \geq \phi_2(g(i, t)) + \frac{\alpha}{\beta} \phi_1(1) \cos(\theta)$, we have:

$$\phi_2(g(j, t)) \geq \phi_2(g(i, t)) + \frac{\alpha}{\beta} \phi_1(1) \cos(\theta) (0 \leq \theta \leq \pi) \quad (3)$$

$$\Rightarrow \phi_2(g(j, t)) - \phi_2(g(i, t)) \geq \frac{\alpha}{\beta} \phi_1(1) \cos(\theta) \quad (4)$$

$$\Rightarrow \alpha(\phi_2(g(j, t)) - \phi_2(g(i, t))) \geq \beta \phi_1(1) \cos(\theta) \quad (5)$$

$$\begin{aligned} &\Rightarrow \alpha(\phi_2(g(j, t)) - \phi_2(g(i, t))) \\ &\geq \beta(\phi_1(a_{ki}) - \phi_1(a_{ij})) \cos(\theta) (\forall k, 0 \leq \theta \leq \pi) \end{aligned} \quad (6)$$

$$\Leftrightarrow \varphi(j) > \varphi(i) \quad (7)$$

Now we see that utility function φ is monotonically increasing and the cell that the target is located gives the highest information utility. This completes the proof of our theorem that PMBT is guaranteed to deliver all tracking messages to the target's location given the conditions.

Note that during the tracking process a querying message does not know the real location of the target. We can only use the recorded/estimated target location to compute the projection angle θ . Obviously, the effectiveness of the tracking algorithm is affected by the precision of estimating θ . In the following discussion, we first prove that the estimation error (of θ) is bounded, and then derive the formula to calculate the confidence as the quality measure of the gradient.

Theorem 2. *The error (ε) of calculating the projection angle using the recorded target's position is bounded by $\arcsin(\frac{vt}{D})$, where D is the distance from current position to the recorded target's position, v is the target speed, and t is the time elapse since last encounter. That is, the real projection θ_{real} satisfies: $\theta_{est} - \arcsin(\frac{vt}{D}) \leq \theta_{real} \leq \theta_{est} + \arcsin(\frac{vt}{D})$.*

Proof. As shown in Figure 4, on calculating the utility value from node i to node j , the estimated θ (the angle $\angle jit$) is used. Within time t , the real target must be located in the dashed circle with radii vt , regardless of its mobility model. Therefore, the maximum variation of θ is $\arcsin(\frac{vt}{D})$. Hence $\theta_{est} - \arcsin(\frac{vt}{D}) \leq \theta_{real} \leq \theta_{est} + \arcsin(\frac{vt}{D})$.

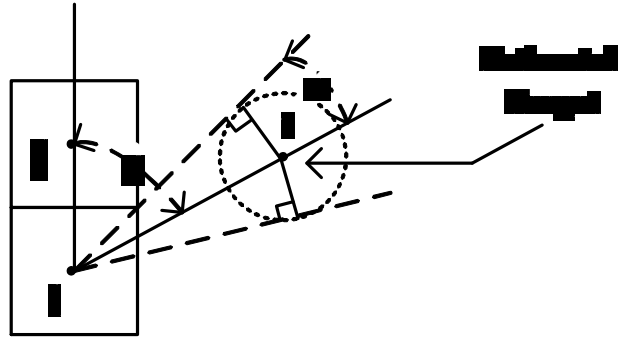


Figure 4: Error bound for projection angle

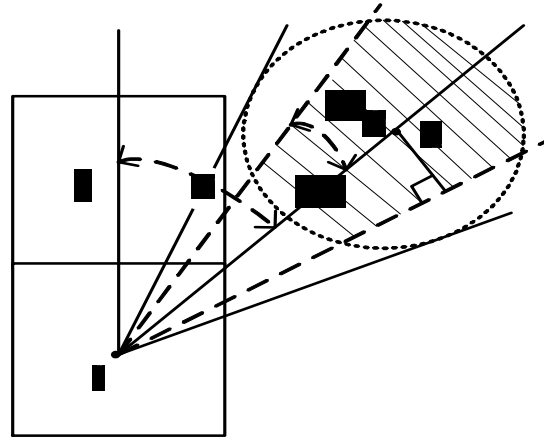


Figure 5: Probability of a predefined projection error

Corollary. Given an error tolerance ε_0 , the confidence of projection defined as $P(\theta - \varepsilon_0 < \theta < \theta + \varepsilon_0)$ for random waypoint mobility model, can be calculated as follows:

$$P(0 < \varepsilon \leq \varepsilon_0) = \frac{r_m^2 + 1}{2r_m^4 \pi} \left[r_m^2 \pi + 4D^2 (\sin \varepsilon_0)^2 \arccos\left(\frac{D \sin \varepsilon_0}{r_m}\right) - 2D \sin \varepsilon_0 \left(\frac{r_m^2 - D^2 (\sin \varepsilon_0)^2}{r_m^2} \right)^{\frac{m}{2}} + r_m^2 \arcsin\left(\frac{D \sin \varepsilon_0}{r_m}\right) \right]$$

where $r_m = vt$.

In [22], the node distribution probability density function for random waypoint mobility model in a circle is given in equation 8:

$$f(r, \phi) = f(r) = \frac{-2}{r_m^4 \pi} r^2 + \frac{2}{r_m^2 \pi} \quad (8)$$

where $0 \leq r \leq r_m$. In our case, $r_m = vt$. Given an error bound ε_0 , the probability of the projection angle within the range $P(\theta - \varepsilon_0 < \theta < \theta + \varepsilon_0)$ is the probability of the node located within the shaded area in Figure 5, which we call the confidence. To compute the confidence, we first compute the error probability P_{err} by the following equation 9:

$$P_{\text{err}} = P(\varepsilon > \varepsilon_0) = \int_{D \sin \varepsilon_0}^{r_m} \int_0^{2 \arccos \frac{r}{r_m}} f(r) \cdot r \cdot \phi d\phi dr \quad (9)$$

The result of the integration is:

$$P_{\text{err}} = \frac{r_m^2 - 1}{2r_m^4 \pi} [r_m^2 \pi - 4D^2 (\sin \varepsilon_0)^2 \arccos\left(\frac{D \sin \varepsilon_0}{r_m}\right) + 2D \sin \varepsilon_0 \left(\frac{r_m^2 - D^2 (\sin \varepsilon_0)^2}{r_m^2}\right)^{\frac{m}{2}} - r_m^2 \arcsin\left(\frac{D \sin \varepsilon_0}{r_m}\right)] \quad (10)$$

Accordingly, the confidence is $P(0 < \varepsilon \leq \varepsilon_0) = 1 - P_{\text{err}}$, which is:

$$\begin{aligned} &P(0 < \varepsilon \leq \varepsilon_0) \\ &= \frac{r_m^2 + 1}{2r_m^4 \pi} [r_m^2 \pi + 4D^2 (\sin \varepsilon_0)^2 \arccos\left(\frac{D \sin \varepsilon_0}{r_m}\right) - 2D \sin \varepsilon_0 \left(\frac{r_m^2 - D^2 (\sin \varepsilon_0)^2}{r_m^2}\right)^{\frac{m}{2}} + r_m^2 \arcsin\left(\frac{D \sin \varepsilon_0}{r_m}\right)] \quad (11) \end{aligned}$$

As we can see from this formula, the confidence is a function of D and r_m , which is directly derivable if a node can record the location and timestamp of its last encounter with the target. Thereby it is possible to evaluate the confidence at each step.

Adaptive Tuning of β . Tuning of the coefficients α and β helps to improve the tracking performance. For this purpose, we use the confidence defined in Corollary as the quality measure of the gradient (q_g). The quality measure infers the usability of the gradient. We adaptively tune β by incorporating the quality into our model¹.

We call the model capable of adaptively tuning of α and β the Adaptive-PMBT model, or A-PMBT model. High q_g has positive impact on β and negative impact on α . In the A-PMBT model, α and β are no longer constant, but dependent on q_g . Hence the utility function in the A-PMBT model becomes equation 12:

$$\varphi(i) = \alpha(q_g) \phi_1(a_{ij}) \cos(\theta_{ij}) + \beta(q_g) \phi_2(g(i, t)) \quad (12)$$

We compare the performance of the A-PMBT model with the PMBT model in the simulation section.

In the HMM model, the state is not directly achievable, but variables influenced by the states are achievable. Each state has a probability distribution over the possible output. Therefore the sequence of observations generated by an HMM gives some indications about the sequence of states. For simplifying the PMBT model, we assume the states are directly observable, hence in fact incorporating a regular Markov model. Later we will show that using a loose synchrony mechanism, PMBT can still achieve good performance while keeping the system lightweight. In summary, PMBT is a progressive framework that provides incremental accuracy at each step. That is, each cell forwards the query message to the next cell with the maximized information gain to potentially increase the accuracy.

¹ As α and β are constrained weights for a_{ij} and $g(i, t)$, tuning of β alone is sufficient for our model.

3 Distributed Implementation

Implementation of PMBT needs to consider 1) how data is coordinated in cell layer, and 2) how to forward tracking messages efficiently. Moreover, as we show in the following discussion, our design of soft-state loose synchrony for each cell and polite gossip message forwarding further reduces message overhead to keep the system lightweight.

Note that each cell only holds the information locally, without knowing the neighboring cell's information. To select next move, the cell may ask neighboring cells for their utility information before making a decision. However, this operation not only requires cell synchronization over the data within each cell but also introduces large overhead. To keep the model lightweight, we provide an alternative distributed implementation of message forwarding-Polite Gossip and soft state strategy in the following discussion.

3.1 Loose Synchrony

To maintain the data consistency in a cell, previous mobile tracking frameworks such as VINESTALK [6], GLS [8], and HGRID[19], used strict synchronization: nodes constantly exchange information with neighboring nodes to keep synchronization. This causes extra overhead and consumes unnecessary network bandwidth when all nodes are mobile. We observe that such strict synchronization for cells may not be necessary. Considering the constant changing of topology in our model, the loose synchrony approach is most appropriate because it avoids extra traffic in high mobility circumstances. In PMBT, we loosely synchronize the state of nodes in each cell through snooping the target event by utilizing the nature of wireless broadcast communication. In our implementation, nodes in a cell hold different gradients and utility information that are not exchanged. Markovian transition information used in constructing utilities are cell-based and shared among nodes in a cell. More specifically, the operations of PMBT are: 1) when a new node enters a cell, it picks up the Markov information in this cell by sending a DATA REQUEST message. 2) when a node leaves a cell, the old information associated are simply dropped. 3) when a query comes, the node with maximum utility in the cell decides the next move and forwards the message with "Polite Gossip mechanism", which is described in the following subsection.

Of course, such lightweight implementation trade-offs the possibility of failures. Transient data loss will occur when all the nodes holding the shared information move out of a cell, while those newly joined nodes have no clue. In this case, the nodes simply forward the message to a randomly selected neighboring cell. The reasons of a message failure could be: a message is forwarded to an empty cell or a message is lost due to collisions.

3.2 Polite Gossip Forwarding

Our implementation of PMBT does not need to select a leader node in a cell to response

a query. We make all nodes in a cell behave uniformly as a single node by employing the following polite gossip mechanism. In polite gossip, a node suppresses sending a message on hearing the same message sent by other nodes within the same cell: if during the backoff period, a node hears any other node sending the same message, it drops the message from the sending queue.

To realize polite gossip forwarding effectively, we use two types of backoff timer: a default CSMA backoff timer and an application layer backoff timer. These two timers perform different functions in our model: the default CSMA backoff timer is used for reducing collisions between cells, and the application layer backoff timer is used for suppressing the sending of duplicated messages in a cell. We set the application layer backoff timer of each node inside a cell to be proportional to the node's utility function. Therefore, nodes with high utility functions are more likely to trigger the response messages. The implementation of this scheme nicely integrates PMBT and message forwarding, and is lightweight and scalable.

The application layer backoff timer should be carefully designed to reduce message duplications and conflicts in forwarding. Let T_{app} denote the application layer maximum backoff time and T_{msg} be the time to send a message. If there are n nodes within the same cell, the joint probability of backoff timer for n nodes in the cell $p(t_1, t_2, \dots, t_n)$ can be written in conditional probability in equation 13:

$$p(t_1, t_2, \dots, t_n) = p(t_1/t_2, t_3, \dots, t_n) * p(t_2, t_3, \dots, t_n) \quad (13)$$

The probability that the rest $n - 1$ nodes can hear the first node's transmission and disable their own transmission attempts is:

$$P = \int_0^{T_{app}-T_{msg}} \int_{t_1+T_{msg}}^{T_{app}} \dots \int_{t_1+T_{msg}}^{T_{app}} p(t_1, t_2, \dots, t_n) \dots dt_2, dt_1 \quad (14)$$

The result of this integration is:

$$P = \left(1 - \frac{T_{msg}}{T_{app}}\right)^n \quad (15)$$

Using this formula, we can set T_{app} appropriately according to system requirements.

For example, assuming there are two nodes in a cell on average and requiring that the duplication of sending is less than 20%, the application layer backoff is:

$$T_{app} \approx 10 * T_{msg} \quad (16)$$

In our implementation of polite-gossip forwarding, we use implicit acknowledgment. When a node that transmits a message to the next cell does not hear this message forwarding further by that cell, it will resend its message. This is a best effort mechanism and it fails if the transmitter node leaves its cell before it had a chance to resend the message.

4 Simulations

In this section, we simulate the above protocols in a 2D discrete WSN grid and describe its effectiveness and robustness properties. In the simulations we consider a map of 10×10 discrete cells, with size $100\text{m} \times 100\text{m}$. Each node is aware of the location (x,y) , and is assigned to some cell initially. Before evaluating and comparing the effectiveness and efficiency of the proposed tracking algorithms, let us first verify the underlying cell abstraction framework itself. Simulations are prototyped/written in Java program.

We consider the random waypoint mobility model: a node selects a random point which is uniformly distributed in the field and moves toward the destination at a certain speed. When the node reaches the destination, it rests for a certain period of time and selects the next point. The performance of the framework is closely related to the mobility process as a cell's behaviors are simulated by the physical nodes in the cell that are loosely synchronized.

In other words, if all the nodes move out of a cell (i.e., the cell becomes empty), the cell may simply lose the information, leading to a potential failure. The analysis of cell failure can be used to quantify the usability and scalability of the framework, depending on the design requirements of the system.

We plot the cell failure probability in 2D as shown in figure 6. The failure probabilities decrease with the increase of node density, which is an obvious observation due to the soft-state implementation. For random waypoint mobility model, cells closer to borders are more likely to become empty. With 400 mobile nodes in the field (4 nodes/cell in average), the average failure probability is below 4%; if we increase the node density to 6 nodes/cell, the average failure probability is below 1%. This implies that given a reasonable dense mobile network, the probability of a cell without any node can be negligible.

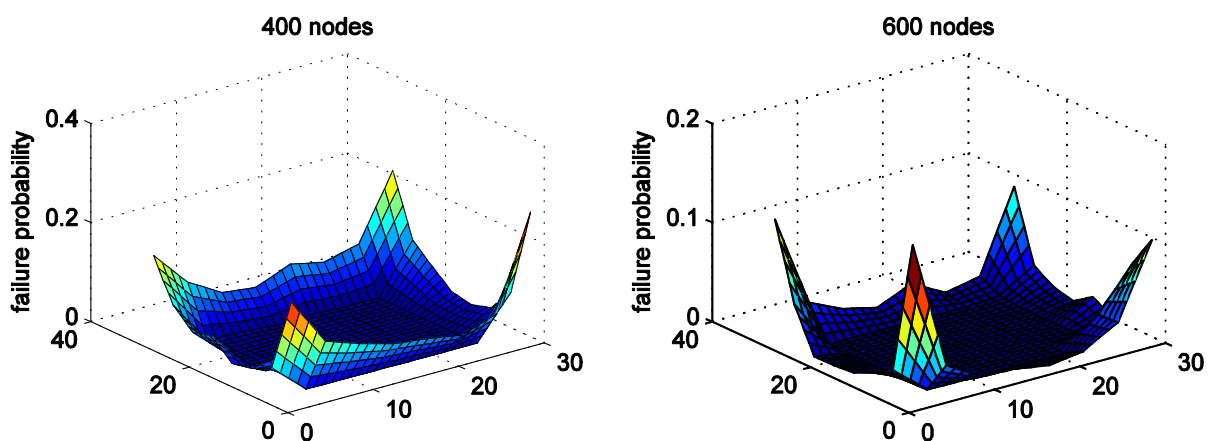


Figure 6: Cell empty probability distribution, random waypoint model

In the following discussion, we present results showing the performance of the discussed

algorithms. The target models we have simulated include a ROAD model where the target's movement is constraint to roads, and a FIELD model where the target is free to move toward any direction. In both experiments, we assume the detecting nodes are free to be mobile, and nodes have already been trained with some initial data. The simulations both in the ROAD model and the FIELD model use a decay function $g(t) = e^{-t}$ with weight factor $\alpha = 0.5$ and $\beta = 0.5$. In both experiments, we use unicast principle: a querying message is sent to the cell with maximum information utility.

To compare their performance, we implement all three models and study the following metrics under different settings. The tracking success rate measures the percentage of tracking messages that are delivered from the initiator to the target. The average path length measures the number of hops a querying message passes before reaching the target. Since the average path length cannot tell the efficiency of different models, we introduce a new metric named average stretch factor to measure distance – sensitivity. Distance-sensitivity for a tracking implies that the cost of a tracking message should be at most a constant factor s of the distance d to the event of interest in the network [21]. Here the constant factor s is what we call stretch factor, and its value can be achieved by calculating the ratio of the path length(in hops) to the distance(in hops) between the tracking node and the target.

4.1 ROAD Model

In this model, we presume the target moves along the road at a constant speed 10 m/s and the stationary track node is located in up-left coner as shown in Figure 7. 300 nodes are randomly located in the field initially and move in random waypoint model. Although the PMBT model is able to predict the trajectory of the target along the road, nonsuccessive detections or temporary node failures will lead to performance degenerations, i.e., the tracking path may become much longer.

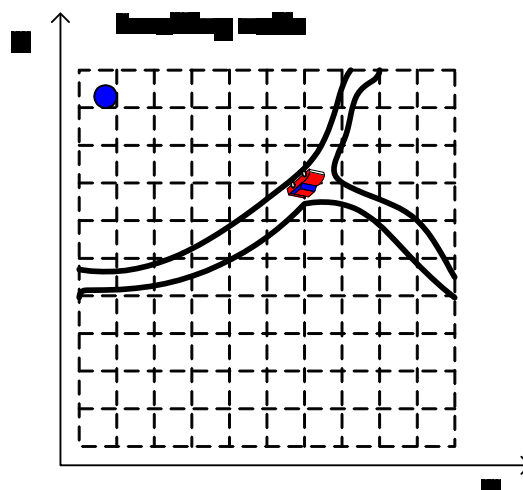


Figure 7: ROAD model scenario.

Figure 8 shows tracking paths from the tracking node to the target for a particular snapshot using different models. As we can see from the figure, the path is usually longer

using the HMM model than using the other two. That is because the cells faraway from the road are absent of any prior knowledge after the training process, making the first few steps relatively blind. However, after reaching the road, it is able to catch the real target movement trace. On the other hand, the Gradient model usually can pick a short path, but it is not sensitive to the movement trail of a target. At any moment, the gradients are determined by the mobile nodes that might not be perfectly inverse to the distance to the target (as we can see from Figure 2, the contour are not perfect circles). That is why we see that tracking paths cross the road instead of simply following the road. The PMBT model (Trust model in the figure) integrates these two models: at the beginning, the path is comparable to the Gradient model; and later it is comparable to the HMM model. As we have stated, PMBT model reaps the advantages of both models, but alleviates their limitations.

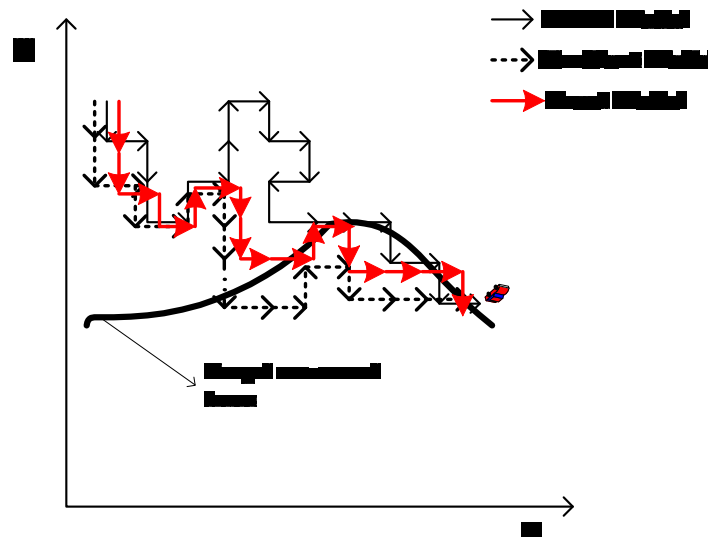


Figure 8: A snapshot of tracking paths for three models.

Table 1 lists the average path length (in hops) and average stretch factor for all models. The tracking node is located at the left corner of the road. We set the buffer size to be 100 (messages), and observe that the tracking success rate is more than 95% for both the HMM model and the PMBT models (both PMBT and A-PMBT). Cell emptiness, forwarding loop, or queue overloading are the main factors to cause failures. PMBT, A-PMBT, and the HMM model have shorter average path lengths and average stretch factors than the Gradient model. This is mainly because the transition probability dominates the message forwarding, making the HMM model and the PMBT model more effective. A-PMBT slightly out performs PMBT owing to the adaptive tuning of parameters. In the simulation we let the tracking node locate on the road that is the best case for HMM; whereas if the tracking node is far away from the road, HMM becomes less efficient to catch the target due to the lack of information around the tracking node.

Table 1: Simulation results for ROAD Model

	HMM	Gradient	PMBT	A-PMBT
Average Path Length(hops)	8.1	14.9	7.8	7.7
Average Stretch Factor	2.2	3.8	2.1	2.1
Success Rate	≈ 95%	90%	96%	97%

4.2 FIELD Model

In this model, we assume the target moves freely in the field; nodes in the communication range can detect the target and update their records correspondingly. Again, all nodes move according to the random waypoint model. Table 2 shows the simulation results in the FIELD model. A-PMBT, PMBT, and the Gradient model perform similarly in terms of the path length and success rate. Both the PMBT models and the Gradient model show better performance than the HMM model due to their partial capability to capture the target's trails. As the target moves in a random model, HMM fails to capture the target efficiently. Tuning of parameters improves the usability of gradients, thereby provides small improvements for the A-PMBT model. Overall, both PMBT and A-PMBT illustrate better performance and resiliency in tracking, and A-PMBT shows best performance among these models.

Table 2: Simulation results for FIELD Model

	HMM	Gradient	PMBT	A-PMBT
Average Path Length(hops)	15.3	10	9.4	9.2
Average Stretch Factor	4.7	3.1	3.0	2.9
Success Rate	80%	91%	96%	97%

We find that for GLS [8] (which hashes location information to the network at different levels), the success rate is above 99% with the same network settings. However, GLS has much higher maintenance overhead as shown in Figure 9. GLS generates three types of protocol packets: HELLO packets to advertise the existence of the node to neighboring nodes, periodic location update packets, and location query and reply packets. PMBT protocol includes two types of messages: location querying/replying packets, and information request packets when a mobile node entering a new cell. As a consequence, PMBT is more lightweight with slightly less accuracy.

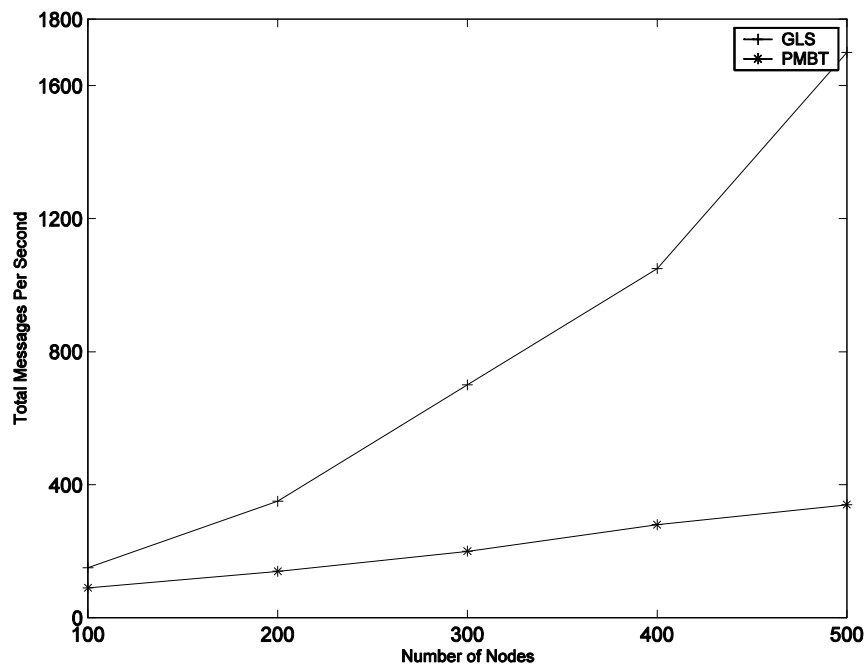


Figure 9: GLS has higher message overhead and is not linearly scalable to the number of nodes; PMBT is more lightweight and linearly scalable to the number of nodes.

4.3 α, β tuning

For PMBT model without adaptive tuning, the initial setting of weights α and β impacts the performance. Changing of the initial weight factor may cause the success rate to increase or decrease. The success rate PMBT model with different weights are shown in Figure 10 for Road environment and Figure 11 for Field environment. In neither mode the line is monotonic, it is low in both ends and high in between. The left end represents Gradient model ($\alpha = 0$), and the right side ($\alpha = 1$) represents an improved Markov model. According to the definition of utility, when $\beta = 0$ it only considers Markov model, whereas PMBT still shows better performance than the Markov model. This is owing to the contribution of projection coefficient, which makes some correction for the transition probability. When $\beta = 1$, this becomes exactly the Gradient model, so the performance of PMBT is close to the Gradient model.

An exact optimal setting is often hard to find due to the dynamics of a system, nevertheless, a range close to optimal setting is more practically achievable. In our case, when $\alpha \in [0.2, 0.8]$ in ROAD model or $\alpha \in [0.1, 0.9]$ in FIELD model, the success rate is close to optimal and does not change much. The size of window is relevant to the selection of decay function and network settings, such as mobility models and node density.

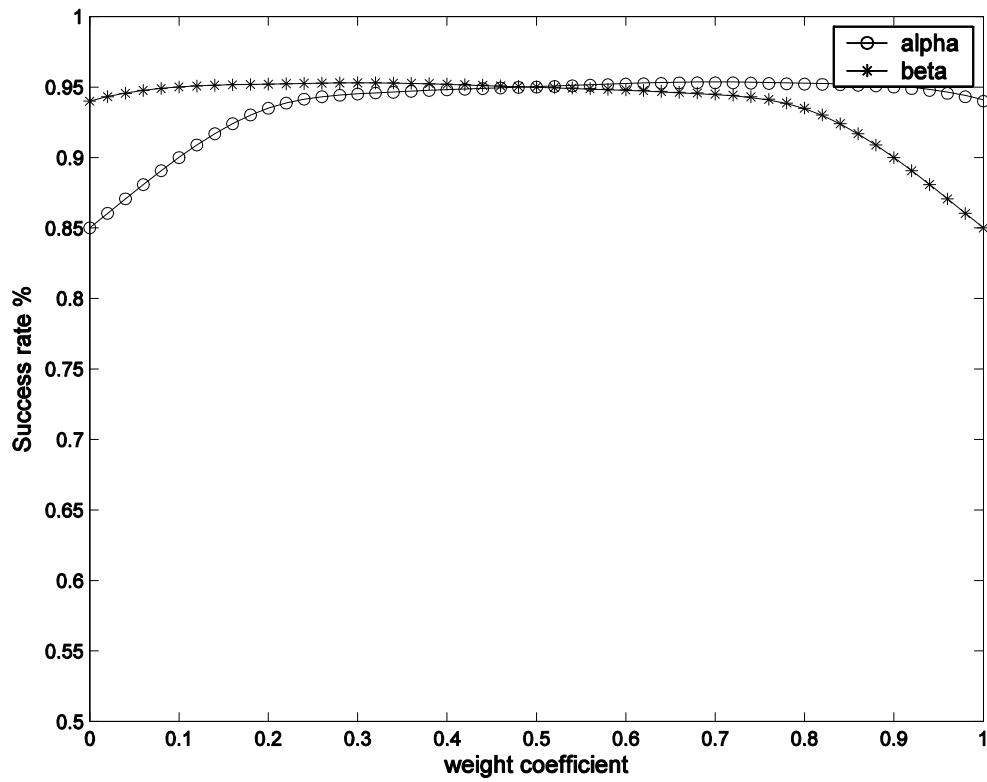


Figure 10: Success rate for ROAD model with different weight coefficients

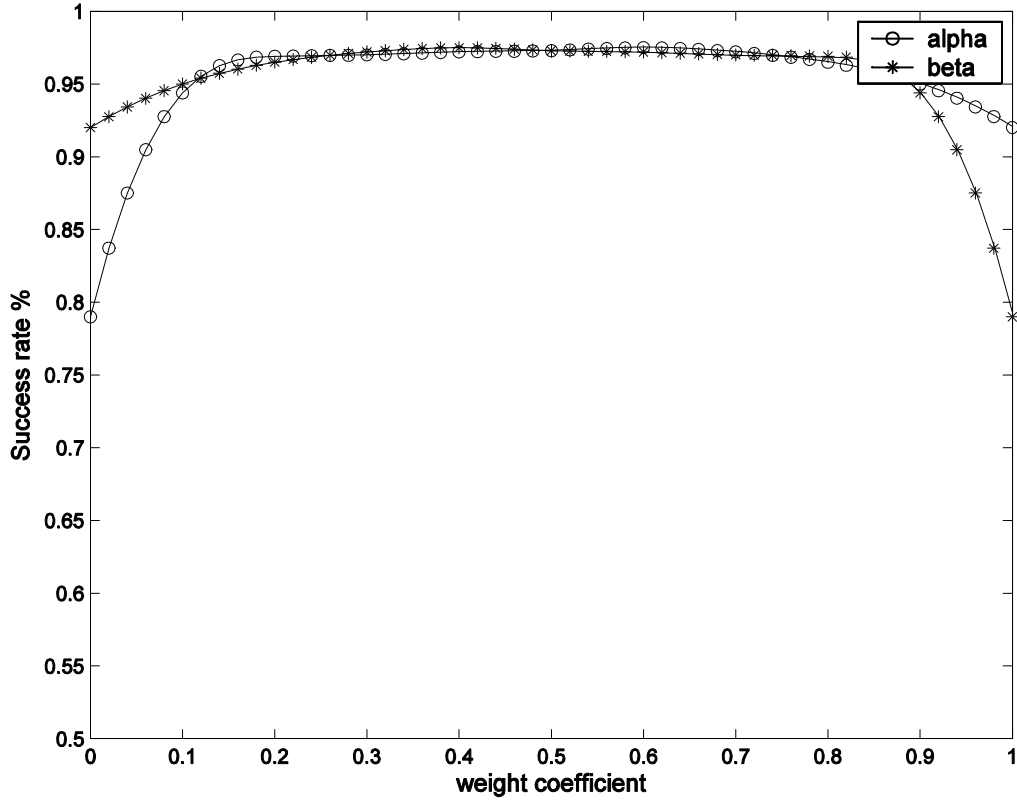


Figure 11: Success rate for FIELD model with different weight coefficients

5 Related Work

Graphical techniques such as HMMs [17, 18] have been investigated and widely applied in target tracking domain. In these approaches, targets move in a rectangular area at each discrete time interval. Cells, associated with states, form an adjacent graphical model. State transitions are only allowed between neighboring cells. HMMs are used to calculate most probable object paths given historical observations. As we mentioned before, HMM models have limitations in nature when applied in WSNs: HMM predicts next move only based on the last location, hence leads to large errors against real trails; Moreover, HMM is more appropriate for offline processing than online tracking.

Dynamic hierarchical/cluster based structures are another category of approaches for mobile target tracking applications [8, 9, 10, 20]). GLS is a scalable and distributed location service structure which divides the global map into hierarchical grids with increased size at high levels. It selects location servers at each level based on least greater ID rules. Queries are forwarded in the same rule by checking location tables. DCTC (Dynamic Convoy Tree-based Collaboration) [10] uses a tree structure called 'Convoy Tree', which is rooted at the target's location and is dynamically adjusted as the target moves. MDQT [9] overlays a virtually static distributed Quad-Tree structure in the field. As a node detects the target, it pushes the event into the tree until the event reaches the root. Hence a frequent update scheme is needed to keep the information up-to-date. In [20], a dynamic clustering framework is proposed for acoustic target tracking in wireless sensor networks. Clusters are dynamically formed and the clusterhead in each cluster is in charge of data collecting, message forwarding and target estimation. All these structure based approaches involve structure reconfiguration via explicit message exchanges at each move for every mobile nodes, hence cost considerable amount of energy. Information driven PMBT model is different from all above in that it does not have a 'tree structure' at all, and thus such coordination cost is reduced.

Gradient based structure-free approaches are also commonly used in routing and tracking applications. GRAB [12] builds and maintains a cost field, providing each node the direction to forward messages. FRESH [14] constructs age gradients for all nodes encountered. Messages are forwarded anchor by anchor that is discovered by broadcast-based searching at each step. Both FRESH and GRAB protocols aim to limit the radius of broadcast to save energy, whereas broadcast is still costly compared with PMBT. GraDrive [13] is also a gradient driven model for dynamic target tracking applications. GraDrive integrates per-node prediction with global collaborative prediction to estimate the location of a target. Although GraDrive provides a shortest path to the target, it requires a global view of the entire dataset, which is often impossible for a fully distributed network. Furthermore, it is also not energy efficient due to the communication overhead for collecting data to the base station.

6 Conclusion

In this paper, we proposed the PMBT model, a generic tracking framework for online

tracking applications, by integrating a Gradient model of the target's proximity and an online statistic Markov model of the target's likely direction. PMBT is a probabilistic online tracking algorithm that computes information utilities at each step, and then chooses the next step toward the target based on the maximum expected utility. PMBT achieves $O(1)$ updating cost with comparable performance as deterministic protocols such as GLS. The A-PMBT model, a PMBT model capable of tuning the weights adaptively based on the evaluation of the gradient quality, provides a slight performance improvement. Both analysis and evaluation on simulations demonstrated that PMBT can significantly improve the performance and reduce the overall transmission cost compared to using the Markov model or Gradient model alone.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, pages 102–114, 2002. <http://dx.doi.org/10.1109/MCOM.2002.1024422>
- [2] I. Chlamtac, M. Conti, and J. Liu. Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Network Journal*, Vol.1, No.1, pp. 13–64, 2003. [http://dx.doi.org/10.1016/S1570-8705\(03\)00013-1](http://dx.doi.org/10.1016/S1570-8705(03)00013-1)
- [3] G. Nelson, K. Blankespoor, and M. Raibert. Walking bigdog: Insights and challenges from legged robotics. *Journal of biomechanics*, 39(1):S360, 2006.
- [4] J. Reich, V. Misra, and D. Rubenstein. Roomba madnet: a mobile ad-hoc delay tolerant network testbed. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(1):68–70, 2008. <http://dx.doi.org/10.1145/1374512.1374536>
- [5] S. M. Das, H. Pucha, and Y. Charlie Hu. On the scalability of rendezvous-based location services for geographic wireless ad hoc routing. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(13):3693–3714, 2007. <http://dx.doi.org/10.1016/j.comnet.2007.03.006>
- [6] T. Nolte and N. Lynch. A virtual-node based tracking algorithm for mobile networks. In *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS)*, page 1, 2007. <http://dx.doi.org/10.1109/ICDCS.2007.82>
- [7] R. R. Brooks, P. Ramanathan, and A. M Sayeed. Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8):1163–1171, 2003. <http://dx.doi.org/10.1109/JPROC.2003.814923>
- [8] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, 2000. <http://dx.doi.org/10.1145/345910.345931>
- [9] X. Lu and M. Demirbas. Writing on water, a lightweight tracking framework for mobile ad hoc networks. *Fifth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MAHSS)*, 2008. <http://dx.doi.org/10.1109/MAHSS.2008.4660072>
- [10] W. Zhang and G. Cao. Dctc: Dynamic convoy tree-based collaboration for target

- tracking in sensor networks. *IEEE Transactions on Wireless Communication*, 3(5):1689–1701, 2004. <http://dx.doi.org/10.1109/TWC.2004.83344>
- [11] C. Westphal. Little tom thumb went straight home. In *Proc. 26th IEEE International Conference on Computer Communications.*, 2007. <http://dx.doi.org/10.1109/INFCOM.2007.215>
- [12] F. Ye, G. Zhong, S. Lu, and L. Zhang. Gradient broadcast: a robust data delivery protocol for large scale sensor networks. *ACM Wireless Networks (WINET)*, 11(3):285–298, 2005. <http://dx.doi.org/10.1007/s11276-005-6612-9>
- [13] Q. Zhang, G. E. Sobelman, and T. He. Gradient-driven target acquisition in mobile wireless sensor networks. *MSN*, pages 365–376, 2006. http://dx.doi.org/10.1007/11943952_31 Key: citeulike:3224439
- [14] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *Proc ACM International Symposium on Mobile Ad Hoc Networking and Computing* , pages 257–266, June, 2003. <http://dx.doi.org/10.1145/778415.778446>
- [15] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: a geographic hash table for data-centric storage. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 78–87, 2002. <http://dx.doi.org/10.1.1.1.4507>
- [16] J. Faruque, K. Psounis, and A. Helmy. Analysis of gradient-based routing protocols in sensor networks. *IEEE/ACM Int'l Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2005. http://dx.doi.org/10.1007/11502593_21
- [17] C. C. Ke, J. G. Herrero, and J. Llinas. Comparative analysis of alternative ground target tracking techniques. *Proceedings of the Third International Conference on Information Fusion*, 2:WEB5:3–10, 2000. <http://dx.doi.org/10.1109/IFIC.2000.859832>
- [18] F. Martinerie. Data fusion and tracking using hmms in a distributed sensor networks. *IEEE Trans. on Aerospace and Electronic Systems*, 33(1):11–28, 1997. <http://dx.doi.org/10.1109/7.570704>
- [19] S. Philip. Scalable Location Management for Geographic Routing in Mobile Ad hoc Networks. PhD thesis, University at Buffalo, SUNY, 2005.
- [20] W. Chen, J. C. Hou, and L. Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transaction on Mobile Computing*, 3(3):258–271, 2004. <http://dx.doi.org/10.1109/TMC.2004.22>
- [21] M. Demirbas, X.Lu, and P.Singla. An in-network querying framework for wireless sensor networks. *IEEE Transaction on Parallel and Distributed Computing*, 2009. <http://dx.doi.org/10.1109/TPDS.2008.217>
- [22] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003. <http://dx.doi.org/10.1109/TMC.2003.1233531>

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).