A Holistic Solution to Pursuer-Evader Tracking in Sensor Networks

Xuming Lu, Murat Demirbas, Chunming Qiao
Computer Science and Engineering Department
University at Buffalo, SUNY
Email: xuminglu, demirbas, qiao@cse.buffalo.edu

***Abstract*—In this paper we devise a holistic solution to the pursuer-evader tracking problem taking into account the limitations of the wireless sensor networks (WSNs) as well as the dynamics of both the pursuer and evader. More specifically, we present an optimal strategy for the pursuer to capture the evader despite the delayed and imprecise information available at the pursuer-side. In order to minimize the communication overhead while ensuring capture, we provide an optimal evader-sampling scheme that adjusts the sampling frequency based on the strategies of the pursuer and evader, as well as the distance between the pursuer and evader. We support our adaptive sampling scheme with a just-in-time delivery protocol that publishes the evader's location updates directly to the pursuer, reducing the communication overhead of tracking even further. To further enhance the tracking reliability, we use a two-level design of fault tolerance: 1) a double position advertisement scheme to mask single message losses, and 2) a breadcrumbs-based backup scheme for stabilizing from desynchronization.**

Our simulation results show that the adaptive sampling scheme guides the pursuer to capture the evader effectively, and reduces the communication overhead significantly compared to fixed rate sampling. Our simulation results also show that our two-level fault-tolerance strategy ensures high capture rates even under consecutive message losses.

I. Introduction

The pursuer-evader tracking (PET) problem considers a pursuer trying to capture an evader and has several applications in the surveillance and military fields. Drawing on the control theory and game theory work, a popular approach to solve the PET problem is based on differential games, where a differential motion model is assumed for both the pursuer and evader [1], [3], [8], [11], [17]. The differential game is essentially an infinite perfect-information zero-sum game, for which a saddle-point equilibrium is found by solving Isaac equations. The Homicidal Chauffeur game and the Lady in the Lake problem are typical examples of the differential game approaches [11]. These works, however, assume that the pursuer has perfect information of the evader: information is available without delay at all times, and information is assumed to be precise. This is clearly not possible in distributed wireless sensor networks (WSNs) due to noisy measurements/estimation errors, partially available information, processing/transmission delays, and message losses.

There has also been several works on PET problem using WSNs [3]–[5], [13], [16]. These works take a one-sided view of the problem, and follow an evader-centric approach by ignoring the dynamics of the pursuer. In this approach, a tree-like structure is maintained continuously to be rooted at the evader, and the movement of the evader causes updates and restructuring of the tree. In other words, this approach provides a decentralized location directory service, and does not take the pursuer strategy or the pursuer location much into account while doing so. The state of the pursuer affects neither the information advertisement nor the reorganization of the tree structure. However, the pursuer is also a big part of the PET problem and the pursuer strategy can influence the way the location queries should be executed. For example, the pursuer location directly influences the frequency with which to send the evader location updates and the latency deadlines that should be satisfied by these updates.

Our contributions. In this paper we devise a holistic solution to the PET problem taking into account the dynamics of both the pursuer and evader as well as the limitations of WSNs. Our main contributions are as follows:

1) We solve the PET problem through exploiting geometry and provide an optimal capturing strategy for the pursuer despite the delayed and imprecise information available at the pursuer-side. We give an upper bound on the uncertainty and delay for which capture is possible.
2) We devise an adaptive evader-sampling scheme that minimizes the communication overhead while ensuring capture. To achieve this goal, our sampling scheme adjusts the sampling frequency based on the strategies of the pursuer and evader, as well as the distance between the pursuer and evader.
3) We provide a just-in-time delivery framework (JIT) that supports our adaptive sampling scheme. JIT does not rely on any structure overlay, and publishes the evader's location updates directly to the pursuer, hence reducing the communication overhead of tracking even further. JIT is designed such that the next sampling information from the evader hits/intercepts the pursuer just-in-time when the pursuer needs to make the next decision.
4) We achieve the tracking reliability through a two-level fault tolerance design. We use a *double position advertisement* scheme to mask single message losses without performance penalty, and a self-cleaning *coma* scheme to stabilize from the loss of synchronization between the pursuer and evader upon consecutive message losses.

Overview of our approach. In Section IV we first investigate the optimal strategy for the pursuer to minimize the distance to the evader if perfect information were available at the pursuer-side. We prove in *Theorem 1* that for each sampling interval t, the optimal strategy for a pursuer to minimize the distance to the evader is to move directly toward the evader's expected location in t. As an estimation error is unavoidable in WSN environments, we then investigate the impact of the estimation error and show that it is distance sensitive in *Theorem 2*. To evaluate the usability of an estimation with certain confidence, we provide a quantitative measure of uncertainty in *Theorem 3*.

In order to relax the perfect information requirements for the optimal strategy, in Section V we present an adaptive sampling rate that minimizes the communication overhead while ensuring capture. Our adaptive sampling is based on the distance between the pursuer and evader (d_{pe}); the closer the distance, the higher the sampling rate. Furthermore, the actual strategies of the pursuer and evader are also evaluated: if the evader takes an action that is greatly deviating from the expected trajectory, the sampling rate increases as follows from *Theorem 4*.

We present our JIT delivery protocol in Section VI to support our adaptive sampling scheme. To further improve tracking reliability, in Section VII, we use a double position advertisement scheme to mask single message losses, and a backup scheme to recover from desynchronization. We present the simulation results in Section VIII.

II. Related Work

Due to its relevance to several applications, the PET problem has been studied extensively both in game/control theory field and in the networking field [1], [9], [10], [14], [17], [20].

In the classical pursuit-evasion games, the states of the game is known to both players since the initial state and motion equation are given and states at any time can be described by mathematical formulations. As the Nash equilibrium for zero-sum differential games is a min-max optimal strategy, it satisfies the following saddle-point equilibrium:

$$J(\gamma_1^*, \gamma_2) \leq J(\gamma_1^*, \gamma_2^*) \leq J(\gamma_1, \gamma_2^*)$$

for all γ_i, where $J()$ is the cost function (for instance, distance or capturing time) and γ_1^*, γ_2^* is the optimal strategy pair. This is equivalent to solve the following condition:

$$V(\theta, t) = min\ max \int J_t(p, e)$$

Under the assumption that value function V is continuously differentiable for θ and t, this can be determined by solving the PDE, which is also called Isaacs equation [1], [11]. The worst case analysis [8] assumes that the evaders have global knowledge of the system and can actively move against pursuers, while pursuers do not know the strategies of the evaders. Different from previous works that assume worst case scenario for evaders, BEAR [17] considers probabilistic models where evaders are considered to be no superior than pursuers. More specifically, BEAR uses a greedy scheme that takes a move that yields the highest probability of containing an evader over all possible moves. In [9], the PET problem is described as a partial information Markov non-zero game with integrating map-learning and pursuit. The pursuers and the evader try to respectively maximize and minimize the probability of capture at the next time instant.

Note that the differential game solution minimizes the maximum cost under worst scenarios as the evader actively takes an action to maximize its cost function. Instead, our approach considers considers average cases, and minimizes the expected capturing cost by following the expected trajectory of the evader, which is a more realistic scenario. Our approach also differs from the traditional differential games in that it is a discretized game that employs the adaptive sampling scheme. Our approach relaxes the demand of perfect information availability at the pursuer-side to that of an intermittently available partial and noisy information.

The PET problem has also has been studied under networked environments. In [3], an optimal solution protecting a linear asset in a sensor network setting has been proposed. The pursuit strategy is formulated under communication constraints such as message loss and packets delay. Apart from the distinctions in the problem definition, our solution differs from [3] in the following: 1) unlike in [3], where PET is considered as a differential game where the equilibrium strategy is used, our solution makes no assumption on the evader model; 2) our adaptive sampling considers the relative distance as well as the dynamics of both the pursuer and evader; 3) instead of indexing the evader advertisement in a structure–*Trail*–in [3], we only require the advertisements in-between the evader and the pursuer. Other works in [15], [16] use centralized periodic updated station to collect and process the data, then forward the decision to pursuers agents. Distributed tracking approaches are presented in [4], [10], [22], where sensor nodes dynamically maintain a "tracking tree" that is always rooted at the evader. The pursuer searches the sensor network until it reaches the tracking tree, and then follows the tree to its root in order to catch the evader.

The PET problem is a special case and an extension of typical target tracking problem in WSNs which mainly have 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 [2], [6], [12], [13], [22]. In structure-free protocols, nodes take local decisions by making predictions over available knowledge to generate a path on demand for tracking messages to be forwarded to the target [7], [18], [19], [21]. PET differs from these tracking problem in that 1) it does not require all nodes in the network to know the evader/target. Instead, only the pursuer needs to know the evader's information; 2) it requires smart pursuit strategy to catch the evader in shortest time, thereby is interested in the evader's mobility pattern.

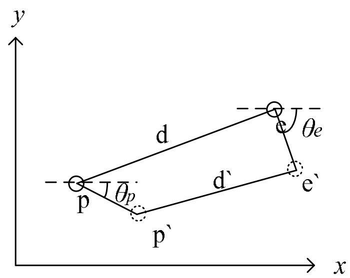

Fig. 1. The Pursuer-Evader Problem

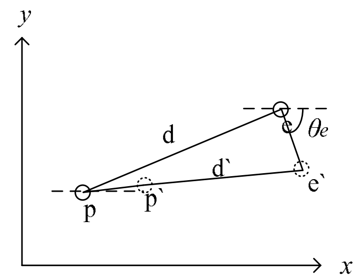

Fig. 2. Optimal pursuit strategy makes p, p', and e' collinear

III. Preliminaries

We consider a pursuer and an evader located in a planar space with wireless sensor nodes covering the entire space. We adopt the following notations to represent the system states. $\mathbf{X}_p = (x_p, y_p)$ denotes the position of the pursuer and $\mathbf{X}_e = (x_e, y_e)$ denotes the position of the evader. The pursuer and evader have the maximum speeds $\mathbf{V} = (v_p, v_e)$, where $v_p > v_e$, and move in the directions $\Theta = (\theta_p, \theta_e)$, respectively. The sampling interval is denoted as t, and it is variable. The new location for the pursuer and evader after t are denoted as $p^{'}$ and $e^{'}$. Between the reception of two consecutive sampling information, the pursuer keeps its direction the same. We say that an evader is captured by the pursuer if $d(X_p, X_e) < \epsilon$, where ϵ is the capture distance.

The underlying control strategy of the evader's movement is unknown to the network. However, in contrast to a non-cooperative evader in differential games, the evader in this paper is actually simulated by an "evader agent" that can forward the detection and direction change of the evader to the pursuer, and is thereby cooperative with the pursuer. Such an evader agent can easily be maintained over the nodes that detect the evader and the state of the evader agent can be handed off to the neighboring set of nodes that detect the evader next as the evader moves.

IV. The Optimal Pursuer Strategy

If differential motion models are known for both the pursuer and evader, an optimal strategy is a saddle-point equilibrium which can be found by solving Isaac equations [11]. These differential game theory formulations assume that the evader always takes an action to minimize the cost function. In contrast, we study the PET problem where the pursuer does not know the strategy of the evader so the game theory formulations are inapplicable. Instead we take a geometric approach.

Theorem 1. *At each sampling interval t, the optimal strategy for a pursuer for minimizing the distance to the evader is to move directly toward the evader's expected position at t.*

Proof: Given the current locations of the pursuer and evader (Figure 1), after t, the new locations are:

$$x_p^{'} = x_p + v_p t cos\theta_p, y_p^{'} = y_p - v_p t sin\theta_p$$
$$x_e^{'} = x_e + v_e t cos\theta_e, y_e^{'} = y_e - v_e t sin\theta_e$$

The initial distance $d = \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2}$ between the pursuer and evader becomes:

$$d^{'} = \sqrt{(x_p' - x_e')^2 + (y_p' - y_e')^2}$$

Without loss of generality, assume $x_p = 0$ and $y_p = 0$, thereby,

$$d^{'2} = (x_e + t(v_p cos\theta_p - v_e cos\theta_e))^2 + (y_e + t(v_e sin\theta_e - v_p sin\theta_p))^2$$

In order to minimize the distance $d^{'}$ upon variable θ_p, we differentiate the equation by θ_p on both sides:

$$\frac{\partial d^{'2}}{\partial \theta_p} = 2t x_e v_p sin\theta_p - 2t y_e v_p cos\theta_p + 2t^2(v_p cos\theta_p - v_e cos\theta_e) v_p sin\theta_p - 2t^2(v_p sin\theta_p - v_e sin\theta_e) v_p sin\theta_p$$

We set $\frac{\partial d^{'2}}{\partial \theta_p} = 0$, and get:

$$\theta_p = arctan(\frac{x_e + v_e t cos\theta_e}{y_e - v_e t sin\theta_e})$$

This indicates that p, p', and e' should stay on a line as shown in Figure 2. Hence, the optimal strategy for the pursuer to minimize the distance to the evader is to move directly toward the evader's expected location (e') in time t. ∎

Note that this result holds only if perfect information of the evader is available to the pursuer. Due to noisy measurements in WSNs, evaluating the uncertainty is needed. In Theorem 2 we show that the estimation uncertainty is distance sensitive. The larger the distance, the smaller the effect of uncertainty on the pursuer's decision.

Theorem 2. *The uncertainty (θ_{pu}) of calculating θ_p is bounded by $2arcsin(\frac{v_e t}{d})$ regardless of the evader's action. That is, θ_{pu} satisfies: $\theta_{pu} \leq 2arcsin(\frac{v_e t}{d})$.*

Proof: As shown in Figure 3, the evader's position is located within an Apoapsis circle[1] with a radius $v_e t$ regardless of its mobility model. Therefore, the uncertainty of estimating θ_p is $\pm arcsin(\frac{v_e t}{d})$, that is, $\theta_{pu} \leq 2arcsin(\frac{v_e t}{d})$. ∎

Theorem 2 indicates that the error of a pursuer's action is inversely proportional to the distance to the evader. If the distance d is small, better estimation is required to keep the same error level.

[1]The farthest points that the evader can reach forms a circle named Apoapsis circle

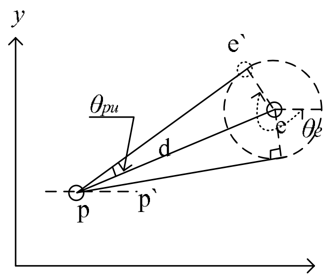

Fig. 3. $\theta_{pu}$is bounded by $arcsin(\frac{v_e t}{d})$

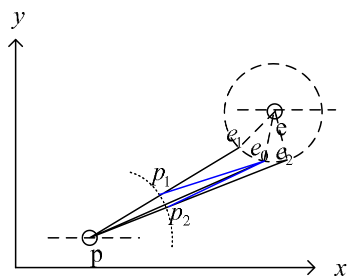

Fig. 4. The maximum error θ_{eu} of the estimating θ_e should be bounded

Theorem 3. *The monotonically decreasing nature of d is maintained if the uncertainty ($\pm\theta_{eu}$) of estimating θ_e is bounded by*
$\theta_{eu} < arcsin(\frac{x_r v_p t cos\theta_e - y_r v_p t + sin\theta_e - \beta}{x_r v_p t sin\theta_e - y_r v_p t cos\theta_e}) < \frac{\pi}{2}$*, where* $x_r = (x_p - x_e)$ *and* $y_r = (y_p - y_e)$*, and* $\beta = (v_p t + x_r cos\theta_e - y_r sin\theta_e)(d - v_e t) - d^2 - (v_p t)^2$.

Proof: Let's assume that the evader moves from e to e_0 in time t. Due to the estimation uncertainty $\pm\theta_{eu}$ of θ_e, the evader can possibly reach e_1 and e_2 in worst cases as shown in Figure 4. p_1 and p_2 are the intersections of pe_1, pe_2 to the pursuer's Apoapsis circle respectively. The locations for e_0, e_1 and e_2 are:

$$x_{e_0} = x_e + v_e t cos\theta_e, y_{e_0} = y_e - v_e t sin\theta_e$$
$$x_{e_1} = x_e + v_e t cos(\theta_e + \theta_{eu}), y_{e_1} = y_e - v_e t sin(\theta_e + \theta_{eu})$$
$$x_{e_2} = x_e + v_e t cos(\theta_e - \theta_{eu}), y_{e_2} = y_e - v_e t sin(\theta_e - \theta_{eu})$$

Accordingly, the projected error for θ_p, $\angle e_1 p e_0$ and $\angle e_0 p e_2$ are given by:

$$\angle e_1 p e_0 = arccos(\frac{|pe_0|^2 + |pe_1|^2 - |e_0 e_1|^2}{2\,|pe_0|\,|pe_1|})$$
$$\angle e_0 p e_2 = arccos(\frac{|pe_0|^2 + |pe_2|^2 - |e_0 e_2|^2}{2\,|pe_0|\,|pe_2|})$$

Hence $|p_1 e_0|$ and $|p_2 e_0|$ can be computed as follows:

$$|p_1 e_0| = \sqrt{|pe_0|^2 + |pp_1|^2 - 2\,|pe_0|\,|pp_1|\,cos(\angle e_1 p e_0)}$$
$$|p_2 e_0| = \sqrt{|pe_0|^2 + |pp_2|^2 - 2\,|pe_0|\,|pp_2|\,cos(\angle e_0 p e_2)}$$

As p_1 and p_2 are the boundary locations that the pursuer can reach, the following condition must be satisfied to keep the monotonicity of d for any t:

$$max(|p_1 e_0|, |p_2 e_0|) < d$$

The relationship between $|p_1 e_0|$ and $|p_2 e_0|$ depends on the actual direction that the evader has taken. If p, e_0, e are collinear, then $|p_1 e_0| = |p_2 e_0|$; otherwise if e_0 is under the line pe as shown in Figure 4, then $|p_1 e_0| > |p_2 e_0|$; and vice versa.

Let's assume $|p_1 e_0| > |p_2 e_0|$ (similar analysis can be used for the other case). This condition is simplified as $|p_1 e_0| < d$. From geometry, $|p_1 e_0|^2$ can be calculated by:

$$|p_1 e_0|^2 = |pp_1|^2 + |pe_0|^2 - 2\,|pp_1|\,|pe_0|\,cos(\angle p_1 p e_0)$$

After substituting $cos(\angle p_1 p e_0)$ and simplifying the inequality we get:

$$|pe_1|\,(v_p t + x_r cos\theta_e - y_r sin\theta_e) + d^2 + (v_p t)^2$$
$$< x_r v_p t(cos\theta_e + cos(\theta_e + \theta_{eu})) - y_r v_p t(sin\theta_e + sin(\theta_e + \theta_{eu}))$$

where $x_r = (x_p - x_e)$ and $y_r = (y_p - y_e)$ are the relative coordinates. Since $|pe_1| \geq |pe| - |ee_1| = d - v_e t$, the equal sign holds if and only if p, e_1 and e are collinear, we can get:

$$(d - v_e t)(v_p t + x_r cos\theta_e - y_r sin\theta_e) + d^2 + (v_p t)^2$$
$$< x_r v_p t(cos\theta_e + cos(\theta_e + \theta_{eu})) - y_r v_p t(sin\theta_e + sin(\theta_e + \theta_{eu}))$$

Since $\theta_{eu} \leq \frac{\pi}{2}$, $0 \leq cos(\theta_{eu}) \leq 1$. Dividing $cos(\theta_{eu})$ on the right side, we get the following more restrictive condition:

$$\beta < x_r v_p t(cos\theta_e + sin\theta_e sin\theta_{eu}) - y_r v_p t(sin\theta_e + cos\theta_e sin\theta_{eu})$$

where $\beta = (v_p t + x_r cos\theta_e - y_r sin\theta_e)(d - v_e t) - d^2 - (v_p t)^2$. Solving this inequation, we conclude that θ_{eu} must satisfy the following condition:

$$\theta_{eu} < arcsin(\frac{x_r v_p t cos\theta_e - y_r v_p t + sin\theta_e - \beta}{x_r v_p t sin\theta_e - y_r v_p t cos\theta_e})$$

■

Theorem 3 provides a quantitative measure of θ_{eu}, which can be employed to evaluate the usability of an estimation with a certain confidence. For instance, a Gaussian distribution with expected θ_e with 3σ variation equals to θ_{eu} suggests that with 95% of confidence, if the pursuer makes a move according to this estimation, the pursuit process will not change the monotonicity of d .

V. Adaptive Sampling Rate

Although, we have shown in Theorem 1 that given a perfect estimation in time t, the optimal pursuit strategy is to move toward the expected destination, in reality the estimation is imperfect; the evader may not actually follow the expected direction. Therefore the monotonically decreasing nature of d may be violated unless the pursuer receives continuous updates about the evader's state. That is, frequent sampling of the evader's state and the delivery of these snapshots to the pursuer are necessary for the pursuer to make correct tracking decisions. On the other hand, continuous and precise snapshots

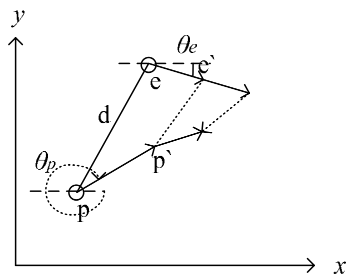

Fig. 5. Adaptive sampling rate

are communication costly for WSNs with limited energy and bandwidth supply.

To solve this dilemma, in this section, we explore an adaptive and variable evader snapshot sampling scheme that satisfies distance sensitive properties; the sampling frequency is higher when the distance d between the pursuer and evader is closer. Moreover, the sampling rate also considers the discrepancy between the expected action and real action: a larger discrepancy leads to more frequent sampling in order to capture the evader effectively.

Theorem 4. *The maximum sampling interval to guarantee the monotonically decreasing nature of d should satisfy (Figure 5):*

$$t_{max} = \frac{x_e(v_e cos\theta_e - v_p cos\theta_p) + y_e(v_p sin\theta_p - v_e sin\theta_e)}{(v_p cos\theta_p - v_e cos\theta_e)^2 + (v_p sin\theta_p - v_e sin\theta_e)^2}$$

This implies that the sampling rate is not only related to the distance d but also related to the strategy of the pursuer and evader.

Proof: As illustrated from *Theorem 1*, the distance of the pursuer and evader at time t is given by:

$$d'^2 = (x_e + t(v_p cos\theta_p - v_e cos\theta_e))^2 + (y_e + t(v_e sin\theta_e - v_p sin\theta_p))^2$$

To find the minimum distance with the given strategy θ_p and θ_e, we differentiate the equation by t:

$$\frac{\partial d'^2}{\partial t} = 2t((v_p cos\theta_p - v_e cos\theta_e)^2 + (v_p sin\theta_p - v_e sin\theta_e)^2) + x_e(v_p cos\theta_p - v_e cos\theta_e) + y_e(v_e sin\theta_e - v_p sin\theta_p)$$

Let $\frac{\partial d'^2}{\partial t} = 0$, we get:

$$t_{max} = \frac{x_e(v_e cos\theta_e - v_p cos\theta_p) + y_e(v_p sin\theta_p - v_e sin\theta_e)}{(v_p cos\theta_p - v_e cos\theta_e)^2 + (v_p sin\theta_p - v_e sin\theta_e)^2}$$

As we have assumed $x_p = 0$ and $y_p = 0$, hence $x_e = x_e - x_p$ and $y_e = y_e - y_p$. After a simple analysis of the equation, we can tell that t_{max} is proportional to the distance d. Consequently, the sampling rate should increase when the distance between pursuer and evader decreases. Since $0 \le t \le \infty$, there are two cases. Case 1, $t_{max} > 0$. In this case, the next sampling should come some time before t_{max}. In other words, t_{max} is the maximum period that pursuer can wait without loss of the optimality. Case 2, $t_{max} \le 0$. This indicates that the current strategy is already not optimal, and an immediate resampling is required. In this way, the adaptive sampling scheme handles the "smart" and "dumb" evaders separately.

Similarly, the minimum distance in t using current strategy is:

$$min(d) = \sqrt{\frac{({x_r}^2 + {y_r}^2)d^2 - (x_e x_r - y_e y_r)^2}{{x_r}^2 + {y_r}^2}}$$

where $x_r = v_p cos\theta_p - v_e cos\theta_e$ and $y_r = v_e sin\theta_e - v_p sin\theta_p$. ∎

The adaptive sampling rate is set such that the monotonically decreasing nature of d is preserved. Theorem 4 implies that the sampling rate is not only related to the distance d but also related to the strategy of the pursuer and evader: the farther the distance and the more favorable the evader strategy, the smaller the sampling rate can be. Therefore, the adaptive sampling scheme can be made more energy efficient than fixed snapshot sampling under these conditions.

VI. Just-In-Time Delivery Protocol

In this section, we describe our just-in-time delivery framework (JIT) for supporting our adaptive sampling scheme. JIT consists of four components: measurement collection, location prediction, decision making, and adaptive sampling. Next we discuss these components briefly.

Sensor nodes detecting the evader collect necessary measurements, such as positions and moving directions, and collaboratively emulate the evader agent. The evader agent maintains a history of these observations $s_1, s_2, ..., s_k$. The state of the evader agent is handed off to the neighboring set of nodes that detect the evader next as the evader moves. This way the evader agent is always co-located with the evader.

In JIT, the distance sensitive adaptive sampling is used to determine how long to delay the next notification to the pursuer, which we call deadline T. The deadline should not only consider the pursuer-evader strategy as we discussed in the previous section, but also take into account the message transmission delay (end-to-end delay). The end-to-end delay is an estimated delay due to channel contention and packet delays occurring at all layers. To this end, an average per hop delay is estimated and then this is extrapolated linearly according to the distance between the evader and pursuer.

When the above calculated sampling interval expires, the evader agent estimates the evader's location at the end of next sampling interval using a prediction algorithm. As location prediction has been extensively studied in the literature, we are not proposing a new algorithm for prediction. This location estimation for the evader is forwarded to the pursuer for the decision making. The evader agent should also estimate the location of the pursuer. This is achieved as follows: The evader agent knows the location of the pursuer initially. When the evader is first detected, the evader agent forwards this detection information to the pursuer's predetermined initial location. Then, the evader agent can determine the next location of the pursuer without any need for communication, because the pursuer uses a deterministic strategy, and the evader-agent is aware of all the inputs it provides to the pursuer side. The implementation of JIT does not rely on any underlying routing protocol.

Therefore, the JIT algorithm for the evader agent is shown in the following table. A node detecting the evader evaluates the message delay that is needed to notify the pursuer (i.e., T_d), and if $T - T_d$ expires, it will predict the evader's location based on the history information (line 4). Then it computes the direction θ_p that the pursuer will take given current θ_e using the formulation in *Theorem 1* (line 5). Furthermore, the sampling deadline T is renewed adaptively as in *Theorem 4* (line 6). All of these information will be forwarded to the pursuer (line 7) for the optimal decision making. Finally the history data, along with the sampling deadline, is recorded at the evader agent (line 9). The algorithm stops when the evader is captured (i.e., $d_{pe} < \epsilon$).

	JIT distributed algorithm
	Input: evader detections, per hop delay **Stop:** d_{pe} is less than predefined distance ϵ
1.	**if** $d_{pe} < \epsilon$
2.	report capture, stop.
3.	**else if** sampling deadline $(T - T_d)$ expires
4.	estimate the evader's location
5.	compute θ_p
6.	recompute sampling deadline T
7.	send θ_e and T to the pursuer
8.	estimate the pursuer's location at next sampling
9.	add new measurements to the history data
10.	**end if**

VII. Fault Tolerance

Our approach to handle reliability is two leveled: masking single message losses and stabilizing from desynchronization. Our fault tolerance design considers a single message loss as an anticipated fault, and masks it without introducing any bad consequences. The second level handles the pursuer-evader desynchronization due consecutive message losses that is not addressed by the masking process. Our reliability improvements are gained at the cost of advertisement overhead.

A. Masking Single Message Losses

Note in our tracking algorithm, the evader agent sends advertisements to the pursuer, and here we assume that no two consecutive advertising messages are lost. In the following we introduce a double position advertisement scheme to enhance the tracking reliability by masking a single message loss.

As the name "double position advertisement" hints, the evader's location is published to two locations instead of a single location. Figure 6 illustrates our double position advertisement scheme. Consider step t_1. An advertisement is sent to p_1 assuming that the last evader advertisement is received at p, and also to $p_1^{'}$ to mask the case that the last evader advertisement is lost at p and the pursuer followed the old strategy to reach $p_1^{'}$. We call p_2 the rally position for $p_1^{'}$. That is, if the pursuer misses a message at t_0 (hence it is at $p_1^{'}$ at t_1), it will move toward p_2 during t_2. Since we have

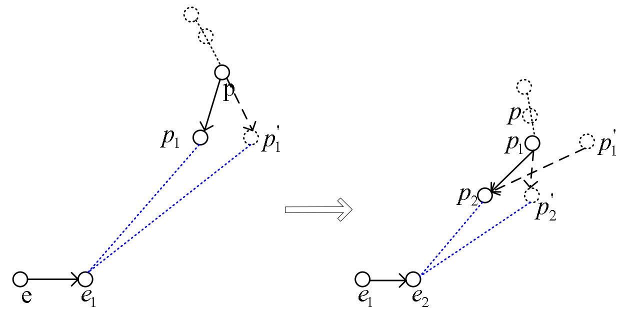

Fig. 6. The figure shows two steps at t_1 and t_2. At each step, the evader agent sends double advertisements to two possible pursuer locations.

assumed that no two consecutive messages will be lost, the advertisement to $p_1^{'}$ will be correctly received, and through this double position advertisement, the pursuer returns back to normal at t_2.

Similarly, the advertisement is published to two locations at each step. Thus, p_2 and $p_2^{'}$ are the advertising locations without/with the loss of advertisement assumption at the next step t_2.

Theorem 5. *The pursuer's speed needs to be increased at least by the following percent for reaching back to the normal operation in the presence of a single message loss:* $\frac{\sqrt{(t_1(cos\theta_{p_0} - cos\theta_{p_1}) - t_2 cos\theta_{p_2})^2 + (t_1(sin\theta_{p_0} - sin\theta_{p_1}) - t_2 sin\theta_{p_2})^2}}{t_2}$

Proof: According to our notation, the locations p_1 and p_2 can be expressed as follows:

$$x_{p_1} = x_p + v_p t_1 cos\theta_{p_1}, y_{p_1} = y_p - v_p t_1 sin\theta_{p_1}$$
$$x_{p_2} = x_{p_1} + v_p t_2 cos\theta_{p_2}, y_{p_2} = y_{p_1} - v_p t_2 sin\theta_{p_2}$$

After replacing x_{p_1} and y_{p_1} in x_{p_2} and y_{p_2}, we get:

$$x_{p_2} = x_p + v_p(t_1 cos\theta_{p_1} + t_2 cos\theta_{p_2})$$
$$y_{p_2} = y_p - v_p(t_1 sin\theta_{p_1} - t_2 sin\theta_{p_2})$$

similarly, the location of $p_1^{'}$ and $p_2^{'}$ are:

$$x_{p_1'} = x_p + v_p t_1 cos\theta_{p_0}$$
$$y_{p_1'} = y_p - v_p t_1 sin\theta_{p_0}$$
$$x_{p_2'} = x_p + v_p cos\theta_{p_1}(t_1 + t_2)$$
$$y_{p_2'} = y_p - v_p sin\theta_{p_1}(t_1 + t_2)$$

If the pursuer does not receive updates at p, it reaches $p_1^{'}$ where it receives the advertisement from e_1. The message contains the position of p_1, which can be used to predict the next location p_2. Then we let the pursuer move from $p_1^{'}$ toward p_2 in order to return back to the normal operation. The distance $d_{p_1' p_2}$ can be calculated by

$$d^2_{p_1' p_2} = v_p^2(t_1(cos\theta_{p_0} - cos\theta_{p_1}) - t_2 cos\theta_{p_2})^2$$
$$+ v_p^2(t_1(sin\theta_{p_0} - sin\theta_{p_1}) - t2 sin\theta_{p_2})^2$$

The lowest speed $(v_p^{'})$ for the pursuer to catch up p_2 is: $v_p^{'} = \frac{d_{p_1 p_2}^{'}}{t_2}$, and thereby $\frac{v_p^{'}}{v_p}$ equals to $\frac{\sqrt{(t_1(cos\theta_{p_0} - cos\theta_{p_1}) - t_2 cos\theta_{p_2})^2 + (t_1(sin\theta_{p_0} - sin\theta_{p_1}) - t_2 sin\theta_{p_2})^2}}{t_2}$

As all the parameters on the right side are readily available, the pursuer can easily compute the speed $v_p^{'}$ at each step. ■

B. Stabilizing from Desynchronization

Using the double position advertisement approach in *Theorem 5*, we avoid the loss of synchronization between the pursuer and evader under the assumption that no two consecutive advertisements are lost. If that assumption breaks at a certain step, the evader agent loses the synchronization to the pursuer, which may further lead to the evader's escape.

Fig. 7. A coma is the nebulous envelope around the nucleus of a comet.

Here, we introduce a backup scheme using the concept of *coma* to solve the desynchronization problem. As shown in Figure 7, a coma[2] is the nebulous envelope around the nucleus of a comet. In our context, a *coma* is a history trace of an evader, which is maintained through forwarding pointers by the relevant nodes, namely by the recent locations of the evader. Each pointer is assigned a *lease* when it is initially generated, so that it can be self-cleaned upon lease expiration. Therefore maintaining the coma does not require any communication cost. Nodes just remember that they are on the evader's trail for the duration of the lease, and each keeps a pointer to the next node on the trail (which can be learned by snooping or during evader agent handoff).

A pursuer agent missing two consecutive messages initiates a resynchronization message containing its own location information, and addresses the message to the last known evader location. This resynchronization message follows the forwarding pointers in the coma hop by hop until it reaches the evader's current position. At this point, the evader sends its location and strategy back to the pursuer. Through this process, the pursuer and evader are resynchronized. If the first resynchronization fails, the pursuer may initiate a second attempt.

A subsequent question to ask is: what should be the length of *coma* for ensuring successful resynchronization? As the resynchronization process immediately happens after the pursuer lost two consecutive messages, the message is sent to the evader location two steps earlier. Another observation is that the sampling period is distance sensitive and generally decreases as the pursuer and evader come closer ($t_1 > t_2 > \cdots > t_n$). Considering the round trip delay t_r for the resynchronization messages, the lease should be set to $2t_r + 2t_1$ for ensuring the second resynchronization attempt to be answered. As both t_1 and t_r are distance sensitive, so is the length of the coma: the larger the distance between the pursuer and evader, the longer the length of the coma.

[2]Photograph courtesy of Wikipedia.

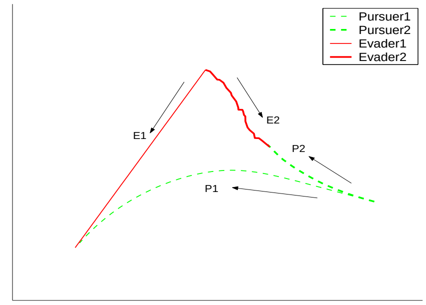

Fig. 8. Pursuer-Evader Trajectories with Perfect Information

VIII. Simulations

In this section, we simulate the JIT protocol in a 2D area using MATLAB. In the simulations we consider an area of 20×20 grid, with $10m \times 10m$ space in each grid cell. A sensor node is assigned to each cell (thus we have 400 sensors in the simulation). Every node is able to detect the evader once the evader moves into its cell. The pursuer's speed is larger than the evader's speed, in our case $\theta_p = 1.3\theta_e$. Regression is used for estimating the next evader position. All the following results are based on same settings and same initial locations, and only allow changes in controlled variables for each experiment.

A. Trajectory Simulation

Figure 8 illustrates pursuer-evader trajectories that follows the optimal strategy with perfect information. The dashed lines are the trajectories of pursuers and solid lines are evaders. Evader1 moves in a straight line and evader2 moves following a direction with random noise. Figure 9 illustrates pursuer-evader trajectories that follows the optimal strategy with information available only at each sampling time using our adaptive sampling scheme. In Figure 9, the same movement patterns as in Figure 8 are used for both evaders. We observe that the trails of pursuers swing from the optimal path due to the intermittent availability of the evaders' information. The decreased precision tradeoffs the demand for continuous sampling and communication.

B. Capture Time vs Estimation Error

We have shown in Theorem 2 and Theorem 3 the effect of the estimation error θ_{eu}. Here, we perform experiments to measure the capture time under varying θ_{eu}. For this purpose, the simulation is set such that at every interval, the evader takes

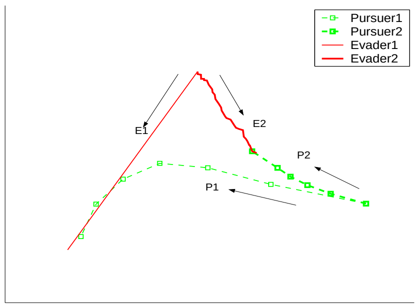

Fig. 9. Pursuer-Evader Trajectories with Adaptive Sampling

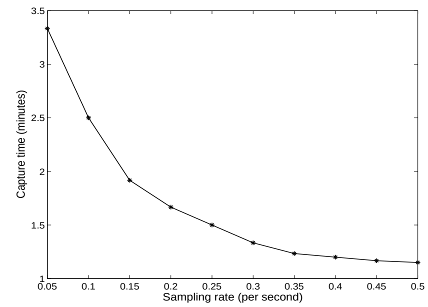

Fig. 11. Increasing sampling rate helps to reduce the average capture time.

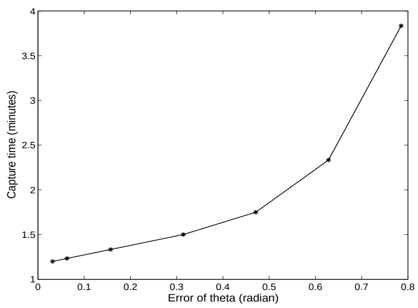

Fig. 10. Reducing estimation error reduces the average capture time.

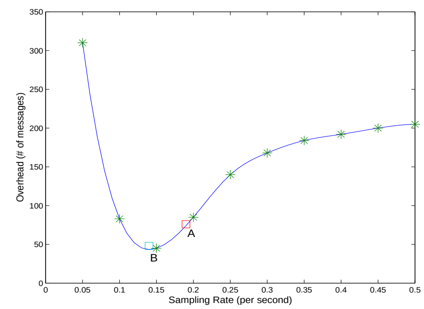

Fig. 12. Overhead shows the dichotomy between the slow and fast sampling

an action θ_e with a random noise within the range $(-\theta_{eu}, \theta_{eu})$. Figure 10 shows that the average capture time increases when θ_{eu} increases. This is because θ_{eu} causes deviated/false decisions for the pursuer. However, beyond a certain threshold the capture time increases exponentially implying the possible loss of capturability.

C. Capture Time vs Sampling Rate

The sampling rate is a tradeoff between the capture time and communication overhead. Higher sampling rate means that more up-to-date information is available at the pursuer, on the other hand it consequently creates more communication overhead. Figure 11 shows the average capture time under different sampling rates (here we refer to fixed sampling rates). The capture time decreases non-linearly as the sampling rate increases, but also the rate of decreasing of the capture time also tends to decrease. The diminishing returns indicate that very high sampling rate is not only costly but also unnecessary. Our adaptive sampling scheme thereby takes advantage of this phenomenon to reduce the overhead while maintaining short catching times.

D. Adaptive Sampling vs Fixed Rate Sampling

Figure 12 shows the overhead of the fixed rate sampling compared to our scheme, using the pursuer1-evader1 case in Figure 8 and 9. The overhead of adaptive sampling for this scenario is 40 messages, and Figure 12 shows the varying overhead for the fixed sampling at different sampling rates. The overhead is calculated by adding up all of the exchanged messages from the start of the simulation until the evader is captured. It is shown in the figure that lowering the sampling rate does not necessarily reduce the overhead. If the sampling rate is greater than a certain threshold (around 0.15 at point B), lowering the sampling rate reduces the overhead; however, if the sampling rate is less than the threshold, the overhead instead increases due to the exponentially increasing of capture time caused by the increased probability of evader escapes.

The overhead in adaptive sampling is always less than using any fixed sampling rate, which verifies its superiority to fixed rate sampling (which ignores the relative distance and strategies of the pursuer and evader). In the figure, point A is the fixed rate that corresponds to the same capture time as the adaptive sampling. In other words, 0.18/second fixed sampling (which corresponds to 80 messages) rate is required to catch the evader in the same time as adaptive sampling. Clearly, the overhead is much lower in adaptive sampling: nearly 50% of communication overhead can be saved in our case. In the figure, although point B has less communication overhead than point A, the capture time of point B is more than that of point A, and also more than that of our adaptive sampling scheme. We also note that the communication overhead at point B is still higher than that of our adaptive sampling scheme.

Note that these results are for the scenario of case1, and at different settings (i.e., initial locations, evader mobility models), the amount of overhead reduction will be different. As the adaptive sampling takes into consideration the pursuer and evader location and their strategy while fixed rate sampling not, more improvements in overhead can be achieved if the relative distance d is larger and the changing of the evader's direction is more likely.

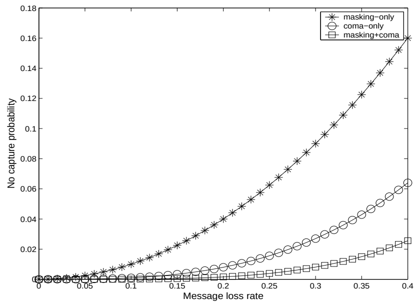

Fig. 13. Our two-level fault tolerance approach improves capturability significantly.

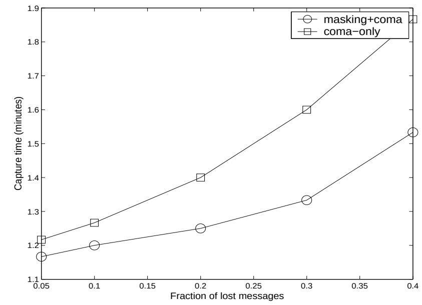

Fig. 14. Capture time increases gracefully under increasing message loss probabilities.

E. Fault Tolerance

We compare the performance under following three different reliability settings:

- masking-only, where only the message loss masking scheme is applied.
- coma-only, where a resynchronization message is sent immediately after detecting a single message loss.
- masking-coma, where a resynchronization message is sent only after the masking scheme fails.

In our simulations, the message loss probability is defined as the loss probability of advertisement/resynchronization messages, not the loss probability at each hop. We use the horizontal axis to denote the fraction of message loss, and the vertical axis for the percentage of no capture probability. Figure 13 shows that significant failure reduction can be achieved by applying masking-coma scheme compared to the masking-only or coma-only scheme. Even with 40% lost messages, the no capture rate with masking-coma is less than 10% in our case.

As a single message loss is masked without penalty in capture time, we only compare coma-only to masking-coma scheme in Figure 14. It shows both capture time increase due to the extra catch-up time and/or resynchronization process. The Coma-only scheme has the longer capture time because of the extra resynchronization delay for every message loss. The Masking-coma scheme covers the single message loss part with little penalty, hence needs less time than the Coma-only scheme. Overall, the masking-coma scheme performs best– with less no-capture-probability and less average capture time– by incorporating both methods.

IX. Conclusion

In this work we investigated the effect of the information available to the pursuer on the outcome of the pursuer-evader tracking and proposed an optimal pursuit strategy to the face of imprecise and delayed information. To reduce the communication overhead, we presented an optimal adaptive sampling scheme that takes in to account the strategies of the pursuer and evader as well as the distance between them. Our fault tolerant design improves the reliability by masking single message losses and recovering from desynchronization due to consecutive message losses. To support our adaptive sampling scheme, we presented a JIT delivery protocol that focuses only on the advertisement between the evader and the pursuer instead of that for the entire network. Unlike previous works, JIT is structure-free and lightweight.

In this paper we assumed that the pursuer can adjust its movement with an infinite granularity. This is unrealistic for environments with obstacles where the pursuer has a maximum turning angle and may not be able to adjust its direction timely as expected. In future work we will investigate the PET problem with mobility constrained pursuers/evaders.

References

[1] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, London and San Diego, 2nd edition, 1999.

[2] 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.

[3] H. Cao, E. Ertin, V. Kulathumani, M. Sridharan, and A. Arora. Differential games in large scale sensor actuator networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 77–84, 2006.

[4] M. Demirbas and A. Arora. A pursuer-evader game for sensor networks. In *Sixth Symposium on Self-Stabilizing Systems(SSS03*, pages 1–16. Springer, 2003.

[5] M. Demirbas, A. Arora, T. Nolte, and N. Lynch. A hierarchy-based fault-local stabilizing algorithm for tracking in sensor networks. *8th International Conference on Principles of Distributed Systems (OPODIS)*, pages 299–315, 2004.

[6] M. Demirbas, X.Lu, and P.Singla. An in-network querying framework for wireless sensor networks. In *To appear in IEEE Transaction on Parallel and Distributed Computing*, 2009.

[7] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters:efficient route discovery in mobile ad hoc networks using encounter ages. *In ACM MobiHoc*, pages 257–266, June, 2003.

[8] L. Guibas, J. Latombe, S. Lavalle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9:471–494, 1999.

[9] J. Hespanha, M. Prandini, and S. Sastry. Probabilistic pursuit-evasion games: A one-step nash approach. In *In proc Conference on Decision and Control*, pages 2272–2277, 2000.

[10] V. Kulathumani, A. Arora, M. Demirbas, and M. Sridharan. Trail: A distance sensitive network protocol for distributed object tracking. In *European conference on Wireless Sensor Networks (EWSN)*, 2007.

[11] J. Lewin. *Differential Games: Theory and Methods for Solving Game Problems With Singular Surfaces*. Springer-Verlag, 1994.

[12] 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 *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, 2000.

[13] 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(MASS)*, 2008.

[14] 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.

[15] S. Oh, S. Russell., and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, 1:735–742 Vol.1, Dec. 2004.

[16] L. Schenato, S. Oh, and S. Sastry. Swarm coordination for pursuit evasion games using sensor networks. In *in Proc. of the International Conference on Robotics and Automation*, 2005.

[17] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662–669, 2002.

[18] C. Westphal. Little tom thumb went straight home. *in Proc. of IEEE INFOCOM*, 2007.

[19] 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.

[20] Y.Zhou and K.Chakrabarty. Distributed mobility management for targte tracking in mobile sensor networks. *Proc. IEEE Trans. Mobile Computing*, 8(8):872–887, 2007.

[21] Q. Zhang, G. E. Sobelman, and T. He. Gradient-driven target acquisition in mobile wireless sensor networks. *MSN*, pages 365–376, 2006.

[22] 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.