Stealthy dissemination in intruder tracking sensor networks

Damla Turgut*, Begümhan Turgut† and Ladislau Bölöni*
*School of EECS, University of Central Florida
4000 Central Florida Avenue, Orlando FL 32816
Email: {turgut,lboloni}@eecs.ucf.edu
†Department of Computer Science, Rutgers University
5110 Frelinghuysen Rd, Piscataway, NJ 08854
Email: bturgut@cs.rutgers.edu

***Abstract*—Many sensor networks are deployed to detect and track intruders. If the existence and location of sensor nodes is disclosed to the opponent, the nodes can be easily disabled or compromised. Wireless transmissions in the presence of the opponent are an important source of disclosure.**

In this paper, we first describe a way to quantify the stealthiness of the sensor node, with a numerical stealthiness metric. Then, we introduce a local model based dissemination protocol, Try and Bounce (TAB) which takes into account stealth considerations while reporting and forwarding observation reports.

In an experimental study comparing TAB to the widely used directed diffusion dissemination protocol, we find that TAB achieves significantly higher stealth for equivalent tracking accuracy, or, alternatively, lower tracking error for equivalent stealth expenditure.

Keywords: sensor networks, intruder tracking, stealth

I. Introduction and motivation

In this work, we consider a sensor network which is used to detect and track intruders in a geographic region. In these *intruder tracking sensor networks* the sensor nodes only record events related to the presence of intruders. These observations are routed or disseminated to the sink, which uses them to track the movement of the intruders on a map of the geographical area considered. The main performance metric of such networks is the real time *tracking error*, the difference between the model maintained by the sink and the real location of the intruders.

The intruders are assumed to belong to a malicious and resourceful adversary, who does not want to be tracked. If the adversary knows the location of the sensor nodes, it can avoid the sensors, find and exploit blind spots, or introduce faked observations. Furthermore, sensor nodes can be easily removed or destroyed if their location is known. A resourceful adversary may even be able to capture and compromise sensor nodes.

We say that the sensor node is *stealthy* if the adversary does not know about its existence. A node is *disclosed* if the adversary can accurately locate the node; this usually allows physical access to the node. Between these two extremes, there might be various levels of stealth. For instance, the adversary might know that with a high probability there are one or more sensor nodes in a certain area, but does not know their exact location.

The stealthiness of sensor nodes depends on many factors. We will differentiate between the factors related to wireless networking as opposed to those related to the physical properties of the nodes.

The physical properties of the nodes such as size, color and mode of deployment can have an important influence on their stealthiness. Large, brightly colored nodes can be easily detected through visual observation. Nodes with large metal components can be detected with a metal detector. If stealth is a requirement, nodes need to be designed such that they blend in with their environment, and need to be deployed through discreet methods. However, once deployed, the nodes can not normally change their color, size or metal content. Thus, the probability of disclosure due to their physical properties will be constant in time.

Another source of disclosure is the wireless transmissions. The sensor nodes can increase their stealth by avoiding transmissions which can be intercepted by the opponent. The nodes need to balance the benefit of transmitting a message with the potential chance of disclosure due to a transmission. We will call *stealthy dissemination* any dissemination or routing algorithm which, considers maintaining the stealth level of the nodes as one of its objectives.

The remainder of this paper is organized as follows. Section II presents related work. Section III gives a formal definition of stealth in sensor networks and suggests methods for its quantification and measurement. Section IV introduces a local model based dissemination protocol, Try and Bounce (TAB) which takes into account stealth considerations while reporting and forwarding observation reports. A simulation study described in Section V compares the tracking accuracy and stealth provided by the TAB protocol to the widely used directed diffusion dissemination protocol. We conclude in Section VI.

II. Related work

A. The topic of stealthiness

The issue of stealth have been addressed at best marginally in the sensor network literature - in form of side comments or paragraph length discussions. To our best knowledge, this is the first paper which proposes a model to quantify the stealthiness of the network node and describes a dissemination algorithm which improves stealthiness compared to other approaches.

Kumar et al. [1] consider a scenario similar to ours and discusses the advantages of stealthiness. This paper however, considers the specific stealthiness as a given rather than a factor which can be influenced by the behavior of the nodes.

The issue of stealthiness, in the sense of the physical location and the discoverability of the sensor is discussed in Cook et al. [2], and used in a decision theoretic approach do decide the positioning of mobile sensor units.

The concept of stealth might also refer to the stealthiness of an attacker. For instance Czarlinska and Kundur [3] consider stealth the ability of an attacker to remain undetected while performing an actuation attack on an event-driven virtual sensor network.

B. Routing in networks with compromised nodes

One of the problems related to the issue of stealth dissemination is the one of routing or disseminating in sensor networks where one or more nodes are compromised (recent work in this direction being Hung et al. [4], Al-Wakeel et al. [5] An and Cam [6]). A probabilistic approach to determine which nodes might be compromised is frequently part of the approaches, for instance in Chen et al. [7]. The overall idea is that the system needs to forward the collected information without relying on the nodes which are compromised with a high likelihood. This is similar to the way in which nodes threatened by intruders, thus in a danger of being disclosed when transmitting, should be avoided when forwarding. However, being under threat is a reversible status, whereas a compromised node is irreversible (although the *suspicion* of being compromised may be lifted under certain circumstances). The approaches frequently include a cryptographic component. INSENS (Deng et al. [8]) is an intrusion tolerant routing algorithm for sensor networks. The goal is both to route around known nodes which are known to be corrupted through techniques such as multipath routing, as well as to defend against various attacks using lightweight security mechanisms.

C. Intruder tracking

Intruder tracking sensor networks has been extensively studied, and the field covers a wide variety of technologies with their specific challenges.

For sensor nodes with plentiful energy resources and guaranteed network connectivity, the scarcest resource is the attention of the sensing device. One example of these approaches is the case when we have a retargetable sensor, such as a directional radar (Horling et al. [9]) or camera tilt and pan angles (Cook et al. [2]). In other papers the assumption is that only a subset of the sensors can be activated simultaneously (eg. Krishnamurty [10]). In all these cases, the challenge is the management of the scarce sensing resources such that the quality of the tracking is maximized.

An alternative scenario is the case when the sensor nodes have limited energy resources. In such systems, nodes need to go through periodical inactive phases to conserve energy and extend the lifetime of the network. The challenge is to assign the active times in such a way that the tracking quality is maintained.

Gui and Mohapatra [11] consider a target tracking sensor network and study the tradeoffs between the power conservation and the quality of surveillance. The surveillance metrics considered in this paper are concerned with the moment of first detection of the intruder in the interest area.

Yan et al. [12] discuss an approach in which nodes self-schedule their active time such that areas with different security requirements are provided differentiated services.

Olariu et al. [13] employs a clustering approach that uses wedges and coronas to create a coordinate system and partition the area around each Aggregation and Forwarding Node (AFN). In this system intruders are not reported to a central location, but are used by mobile nodes trying to avoid threats.

Wang et al. [14] considers the problem of detecting intruders in a network which covers the interest area incompletely and sensors can be heterogeneous in terms of transmission and sensing range. Furthermore, the paper considers the case when the detection of an intruder requires the cooperation of more than one sensor node. The detection probability is mathematically analyzed with respect to various network parameters such as node density and sensing range.

Zou and Chakrabarty [15] consider a target tracking sensor network with mobile units. They aim to improve target tracking by changing the movement patterns of the mobile nodes, at the same time considering the tradeoffs in form of energy expended for movement, potential loss of network connectivity and loss of sensing coverage.

Both the problem of energy conservation, as well as the limited sensing device capacity are orthogonal to the problem of stealth. Whatever the considered setting, stealth is an additional parameter which needs to be optimized for. Fortunately, the compromises between the performance parameters are not necessarily a zero sum game. Techniques which reduce the energy consumption, for instance Kung and Vlah [16], will usually also reduce the number of transmissions, thus improve stealthiness. On the other hand, a stealthy routing algorithm might choose a longer path to route around the intruder nodes and thus require a higher energy consumption.

III. Quantifying stealth

In the following, we will try to quantify the stealth of a node based on a probabilistic interpretation. This allows us to develop a model which describes how the stealthiness of a node evolves in time and in response to transmission events.

We define the *stealth level* $\sigma(t)$ as probability at time t that the node is not disclosed to the opponent. Thus a node

which we are sure is not disclosed will have $\sigma = 1$, while a node about which we know that is is disclosed has $\sigma = 0$. Naturally, it is very difficult to calculate the exact value of stealth. Rather, what we are interested in is the way in which the stealth evolves in response to various actions of the nodes, and in response to the passing of time. This approach is somewhat similar to risk factors in medicine: although the exact probabilities vary from individual to individual, certain behavior patterns increase the probability of disease, while others lower it.

Let us assume that a node was deployed at time point t_0. The deployment of the node was disclosed with a probability p_{deploy}. This implies that $\sigma(t_0) = 1 - p_{\text{deploy}}$.

We assume that the adversary is *interested* in disclosing the node and that it is *not forgetting*. The consequence of these assumptions is that the stealth level of the nodes is non-increasing $t_2 \geq t_1 \rightarrow \sigma(t_2) \leq \sigma(t_1)$.

We assume that at any time there is a constant (albeit low) probability that the node will be disclosed even if the node does not take any action[1]. We denote this *probability of accidental disclosure* in time period $[t, t+1]$ with p_{ad}. Thus, the stealth value of a node which does not take any action is described by:

$$\sigma_0(t) = (1 - p_{\text{deploy}})(1 - p_{\text{ad}})^{t-t_0} \tag{1}$$

We assume that the most important source of loss in stealth are the actions of the sensor, in particular wireless transmissions. If an action i happens at time t_i and has the disclosure probability p_i the stealth level of the node at time t will be:

$$\sigma_0(t) = (1 - p_{\text{deploy}})(1 - p_{\text{ad}})^{t-t_0} \prod_{i, t_i \leq t} (1 - p_i) \tag{2}$$

The disclosure probability associated with a transmission is proportional with the total wireless power received by the opponent, which depends on the transmission power, the length of the transmission and the distance between the sensor and the intruder. In the following, we assume unit-length transmissions. Longer transmissions can be modeled as multiple-unit length transmissions. We assume that there is a certain power level P_G at which the detection is guaranteed, while there is minimal power level P_L at which the detection is not possible. We assume that the P_G is the power at the edge of the nominal transmission range of the node d_{tr}. Assuming a path loss index $n \in [2, 4]$, we have the power at distance d:

$$P(d) = P_G \left(\frac{d_{\text{tr}}}{d} \right)^n \tag{3}$$

Assuming that the probability of detection is linear between power levels P_G and P_L, we find the probability of detection of the transmission by a node actively listening at the correct frequency:

$$p_i = \max \left(0, \frac{min(P(d), P_G) - P_L}{P_G - P_L} \right) \tag{4}$$

This equation assumes that the intruder is actually listening at the moment of the transmission. Naturally, this might not be true. The intruder might not be interested at all in discovering the nodes of the sensor network, its receiving equipment might be currently tuned at a different frequency, or it might be engaged in a transmission. We capture this uncertainty with a probability p_{attn}, which we can assume to be dependent on the intruder. Adding this parameter to the Equation 4 and substituting Equation 3 we obtain:

$$p_i = p_{\text{attn}} \max \left(0, \frac{\min \left(1, \left(\frac{d_{\text{tr}}}{d} \right)^n \right) - \frac{P_L}{P_G}}{1 - \frac{P_L}{P_G}} \right) \tag{5}$$

Based on this formula, a node can estimate the disclosure probability of its own transmission before actually making it. For this, it needs to know the distance to all intruders in its neighborhood. The node can acquire this information either from direct observations or from transmissions of other nodes. In the following we develop a protocol which uses this self-evaluation technique to improve the average stealthiness of its nodes.

IV. Try and bounce, a protocol for stealthy dissemination

The try and bounce (TAB) dissemination protocol is report centric and makes extensive use of a local model of the environment. Instead of focusing on the path to the sink, the protocol focuses on the individual observation reports, the responsibility of the node with respect to the transmission of a report, and the balance between the benefit of the report to the sink versus the cost of stealth occurred by the node when forwarding the report.

In the following we will first describe the path of a report to the sink in the TAB protocol (which would also clarify the reason for its name). Then we will describe in detail the two main components of the protocol: the *maintenance of the local model* and *the decision to transmit*.

A. An example of the path of the report in the TAB protocol

We consider that the node A observed an intruder T and created a report. Now the node A is responsible for the transmission of that report to the sink. Figure 1 illustrates the eight steps through which the report will reach to the sink.

The content of the communicated messages is the *sighting* R and the *path record* P. The sighting is a description of an observation of a threat node, for instance $R = T, 4000, (130, 120)$ which can be read as "intruder T sighted at time $t = 4000$ at location $(130, 120)$". The path record is a triplet of node sets: $\langle Tr, Nt, Np \rangle$, where Tr is the set of traversed nodes, Nt is the set of non-transmitting nodes and Np the set of nodes which have no path to the sink.

[1]In more complex models this probability might not be constant. For instance we can take into account the proximity of intruders, the presence of adversary UAVs, satellites and so on.

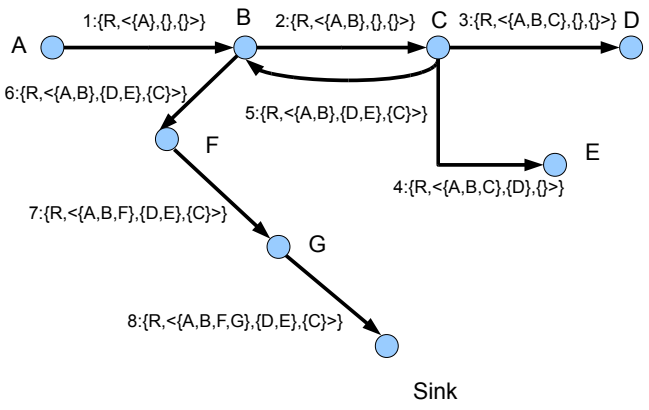

Fig. 1. An example of the path of the report in the TAB protocol.

Step 1: Node A transmits the report $\{R, \langle\{A\}, \{\}, \{\}\rangle\}$ to node B, on its shortest path to the sink. It marks the report in its local model as transmitted but not confirmed, and sets node B responsible for the report. Node B receives the message, introduces the report in its local model and marks itself as responsible for its forwarding.

Step 2: Node B transmits the report with the updated path record $\{R, \langle\{A, B\}, \{\}, \{\}\rangle\}$ to node C, its shortest path to the sink. B marks the report transmitted and marks C as responsible. A overhears B's transmission and marks the report successfully transmitted. From this point on A is not responsible for the tracking of the progress of the report (unless it was explicitly returned).

Step 3: C sends the report to D. B updates it as successfully sent.

Step 4: Node D is *under threat*, it has intruder nodes nearby and although it received the report, it judges that its forwarding is not worth the cost in stealth it would incur, thus it will *not* forward the report. C notices the timeout, which makes it again responsible for the report, and forwards the report to E, which is in its list of the nodes through which the sink is reachable. D overhears this transmission and updates its own copy of the report, unmarking itself as responsible.

Step 5: E does not forward the report either. C is again responsible for the report, but it has run out to possible paths to the sink. Timing out would not work for C, because B had assumed that its responsibility regarding the report has ended. Thus, C will send a *failed transmission* message to B, adding itself to the list of nodes with no path to the sink. The path record will thus be $\langle\{A, B\}, \{D, E\}, \{C\}\rangle$.

Step 6, 7 and 8: B sends the report to F, from where it passes to G and from there to the sink. The senders are updating their models based on their overhearing of the next step (except G which considers its work done as soon as it is transmitted to the sink).

This example illustrated some of the principles behind TAB. In its path to the sink a report might be occasionally bounced back to the sender by nodes refusing to or unable to forward. The decision to forward or not is made locally by the node. This decision is recorded in the path record of the node, but it does not permanently affect the routing tables. A node which has a report is responsible for its transmission, as well as its forwarding in the next hop. If the next hop fails to forward, the responsibility reverts to the sender.

B. Structure and maintenance of the local model

A TAB agent maintains a local model of the environment represented by the triplet $\langle\mathcal{N}, \mathcal{I}, \mathcal{R}\rangle$.

The *set of node models* $\mathcal{N}$ lists the series of sensor nodes known to the agent, whether they are in the transmission range or not and whether they are active, inactive or under threat.

The *set of intruder models* $\mathcal{I}$ contains the list of the intruder nodes currently believed to be in the area of the sensor network, their last known position and potentially other observed properties.

The *set of report models* $\mathcal{R}$ contains a list of reports about intruder nodes to the sink. For each report the model maintains the intruder node, its location, the time when the observation was made and the path record of the report. The model also keeps track whether the node is responsible for the forwarding of the report, or if it has a responsibility in checking its forwarding.

The maintenance of the local model is done by a series of *inferences* triggered every time an agent makes an observation, receives, transmits or overhears a message. In addition, inferences are also triggered by the passage of time. Although the word *inference* would imply a relatively complex process, all the inferences in the TAB agent have been designed to have a $O(1)$ complexity both in time and space. The local model of TAB agent never maintains any historical information. In addition, through the principle of *occlusion*, reports with a newer observation date concerning the same intruder are replacing the older reports. Thus the local model remains compact as the number of reports will not exceed those of the active intruders.

In the following we outline the inference types performed on the local model.

Bookkeeping inferences concern the removal of components from the local model.

> **Occlusion:** A report concerning the same intruder but with a newer observation time discards the previous report regardless of its status. The new report might come either from a direct sighting, from a received message or an overheard message. For instance, a node will discard a report for which it is responsible if it overhears a transmission of a report which occludes it.
>
> **Obsoleting:** Certain types of information are considered expired after an amount of time passes without updates or confirmations. Examples include a node being under threat, location of intruder nodes, and reports. Such information is simply removed when it reaches its validity deadline.

Inferences concerning intruder nodes modify the status of intruder nodes and the threatened status of sensor nodes.

Sighting: A direct sighting of an intruder node updates its model.

Report received: A report received updates the model of the specified intruder node.

Inference from silence: A node which misses its heartbeat message, or does not forward a report is considered to be threatened by an intruder and marked as such.

Inferences concerning nodes: A node is active if it is observing and sensing. We consider a node under threat if it is sensing a threat node - such nodes are not normally a good choice for transmission paths, and, depending on the policies, might not send messages at all. A node is dead if it is unoperational.

Heartbeat: From a heartbeat message, infer that the sending node is alive and under no threat.

Lack of retransmission: If a node did retransmit a message which it should, assume that the node is under threat.

No heartbeat: If a node did not send over a long period of time, make an assumption of a dead node.

Inference from path records: Whenever a node receives or overhears a message with a path record $< Tr, Nt, Np >$, it will mark the nodes in Tr and Np as active non-threatened, while those in Nt as threatened.

Inferences concerning reports: These inferences concern the receipt of the reports. Normally a node is responsible to track the forwarding of reports up to "one hop away".

Report from sighting: The node sighted an intruder and creates a report, assigns to it an empty path record and makes itself responsible for its forwarding.

Report from received message: The node receives a message, and creates a local report with the specified sighting and path record. It makes itself responsible for forwarding. A special case is when the report was a report returned after the next hop failed to forward it, having no path to the sink.

Report transmitted: The node transmits a message and marks the report as transmitted. The node will now be responsible only for the monitoring of the progress of the message to the next hop.

Report progress overheard: The node overhears the forwarding of a previously transmitted report. The report is now marked as successfully transmitted and removed from the model.

Report progress timeout: The node finds that the next hop failed to forward the message. The node is now again responsible for forwarding.

C. The decision to forward a request

At any moment in time, the TAB sensor node will contain in its local model a number of reports for whom it is marked as being "responsible". These reports have been acquired either from the original observations made by the node or have been reported by external nodes. The node does not differentiate between the reports based on the source of the observation. For each report, the node will need to make a decision whether it will transmit it or not, and, in the case of transmission, which next hop will be the target of the transmission.

The simpler question is the target of the transmission. Each node maintains a *dissemination table* $DT = \{n_1, n_2 \dots n_k\}$ which contains the nodes which can serve as possible next hops to the sink, sorted in the order of preference. The first node, n_1 is normally the one on the shortest path to the sink. The next hop n_{next} will be the most preferred node which does not appear in any of the components of the path record.

$$\begin{aligned} n_{next}(\{R, \langle Tr, Nt, Np\rangle\}) = n_i \in N \mid \\ n_i \notin (Tr \cup Nt \cup Np) \ \wedge \\ \nexists j \ \text{ such that } \ j < i \ \wedge \ n_j \notin (Tr \cup Nt \cup Np) \end{aligned} \tag{6}$$

The transmitted report will be of the format $\{R, \langle Tr \cup \{n_i\}, Nt, Np\rangle\}$.

Let us now consider the issue whether the node will transmit. Intuitively, the node needs to balance the stealth loss with the tracking benefit. Ideally, the node would only transmit when the stealth loss is zero, but this is not feasible.

The first step is to estimate the expected stealth loss of a transmission, based on the intruder models $\mathcal{I}$ in the local node.

$$\Delta\sigma = \left(1 - \prod_{i \in I} (1 - p_i)\right) \cdot \sigma \tag{7}$$

where p_i is the detection probability by intruder i calculated using Equation 5 for each of the specific intruder nodes in the area.

The idea is to *cap the stealth loss per intruder per unit of time*. We cannot set a hard threshold because this would lead to a starvation of the node reports in certain situations. Instead, we will maintain a *running average of the stealth loss* $\Delta_{avg}\sigma$ which is updated as follows:

$$\Delta_{avg}\sigma' = \begin{cases} \alpha \cdot \Delta_{avg}\sigma + (1-\alpha)\Delta\sigma & \text{if transmits} \\ \alpha \cdot \Delta_{avg}\sigma & \text{if does not transmit} \end{cases} \tag{8}$$

Having the stealth loss cap S_{cap} as a parameter of the agent, the node will transmit if $\Delta_{avg}\sigma$ for the specific intruder smaller than the stealth loss cap. The overall decision process is summarized in Algorithm 1.

V. Simulation study

A. Objective of the experimental study

Our objective in developing a specific routing protocol for stealthy routing was to obtain a better stealth level of the nodes for an equivalent tracking accuracy. The question is whether moving to a new routing protocol is justified, or whether the existing routing protocols (potentially with adequate parametrization) can achieve the same goal.

Let us, for instance, consider the directed diffusion [17] family of protocols. In these protocols the sink expresses its

input : report models $\mathcal{R}$, threat models $\mathcal{T}$, running average parameter α, stealth loss cap S_{cap}, node stealth level σ
output: transmit decision

for *all reports* $\{R, \langle Tr, Nt, Np\rangle\}$ **do**
  T $\leftarrow$ threat reported by R;
  $\Delta\sigma \leftarrow \left(1 - \prod_{i\in\mathcal{T}}(1-p_i)\right)\cdot\sigma$ **if** $(\Delta_{avg}\sigma(T) > S_{cap}$ **then**
    for $n \in DT$ **do**
      if $n \notin Tr \cup Nt \cup Np$ **then**
        $n_{next} \leftarrow n$ **break**
      end
    end
    `Transmit` $(\{R, \langle Tr \cup \{n_{next}\}, Nt, Np\rangle\}, n_{next})$
    $\Delta_{avg}(T) \leftarrow \alpha \cdot \Delta_{avg}\sigma(T) + (1-\alpha)\Delta\sigma$
  else
    $\Delta_{avg}(T) \leftarrow \alpha \cdot \Delta_{avg}\sigma(T)$
  end
end

Algorithm 1: Algorithm for the routing decision in TAB.

interest on receiving reports about certain types of events. Through the process of interest propagation followed by reinforcement, nodes establish *gradients*, which determine the direction of forwarding of the reports. As we assume that the setup of gradients is done before intruders appear in the area, from the point of view of stealth we are interested only in the normal operation of the directed diffusion model.

In directed diffusion, the original reporting of sightings is guided by the *interval* parameter. If, let us say, the interval parameter is set to 5 seconds, the sensor will send a report on the movement of the intruder in its area every 5 seconds. Thus, the interval parameter allows us to balance the stealth and the tracking error. By setting the interval to a very low value, the tracking accuracy is only limited only by the hop by hop transmission delay. On the other end of the scale, we can achieve arbitrary stealth value, by extending the interval between transmissions.

The objective of our experimental study is to compare the proposed try and bounce protocol with directed diffusion under realistic settings. The goal is not whether a certain accuracy and stealth value can be achieved with the individual protocols, but whether they can provide a better tradeoff between these two performance measures.

B. Experimental settings

For the purpose of the experimental study we have implemented both protocols in the YAES [18] simulation environment. The experimental scenario involves a sensor network with 64 nodes covering an interest area of size 400 by 400 meters. The sensing range and transmission range of the nodes are both assumed to be 50 meters. The scenarios considered span a time interval of 2 hours (7200 seconds), during which a number of intruder nodes traverse the area. The arrival times of the intruders are randomly generated. It is thus possible that at a given moment in the scenario 0, 1 or more intruders are in the area.

Our experiments used four individual protocol instances:
DD-10 - directed diffusion with the interval parameter set to 10 seconds
DD-25 - directed diffusion with the interval parameter set to 25 seconds
TAB-0.001 - try and bounce with the stealth loss cap set to 0.001 stealth units / intruder times second
TAB-0.003 - try and bounce with the stealth loss cap set to 0.001 stealth units / intruder times second

Finally let us provide a definition of the tracking error as it is calculated in our experiments. Intuitively, the tracking error is the distance from the real location of the intruder to the location where the sink believes it to be. The challenge in this case is the handling of the boundary cases. Assuming that the sink is interested in geometric area described by the rectangle R, let us consider an intruder node T, for which the sink has a model $M(T)$. If the sink does not have a model of the the intruder (for instance, if it didn't yet receive a report about it), we will assume that it considers it to be outside the interest rectangle. The calculation of the tracking error $\varepsilon(T, M(T))$ considers the following cases:

- $T \in R \wedge M(T) \in R \Rightarrow \varepsilon(T, M(T)) = dist(T, M(T))$: if both the intruder and the model are inside the interest rectangle, the tracking error is the distance from the model to the intruder.
- $T \in R \wedge M(T) \notin R \Rightarrow \varepsilon(T, M(T)) = dist(T, R)$: if the sink believes the intruder to be outside the interest rectangle but T is inside rectangle R, the tracking error is the distance from T to the closest edge of rectangle R.
- $T \notin R \wedge M(T) \in R \Rightarrow \varepsilon(T, M(T)) = dist(R, M(T))$: if the intruder is not in the interest rectangle, but the sink believes it is, the tracking error is from the model to the closest edge of the rectangle R.
- $T \notin R \wedge M(T) \notin R \Rightarrow \varepsilon(T, M(T)) = 0$: if the intruder is not in the interest rectangle and the sink does not believe it to be in the interest rectangle, the tracking error is zero.

This definition keeps the tracking error continuous as the intruder moves in and out of the interest rectangle. In addition, it respects our intuition that the sink does not care about the intruders outside the area, but penalizes the sink for believing that the intruder is in the interest area while it is not and vice versa.

C. Results: average values

In the first series of experiments we run the simulation scenarios with 5-80 intruder nodes. The same simulation scenario was presented to all four protocol instances. The process was repeated with 10 different random seeds for the movement of the intruder nodes. The average tracking error was measured and averaged over the complete span of the

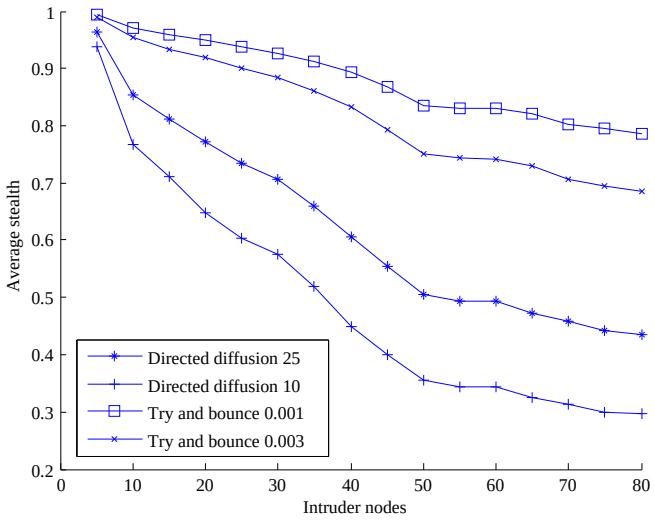

Fig. 2. Stealth averages.

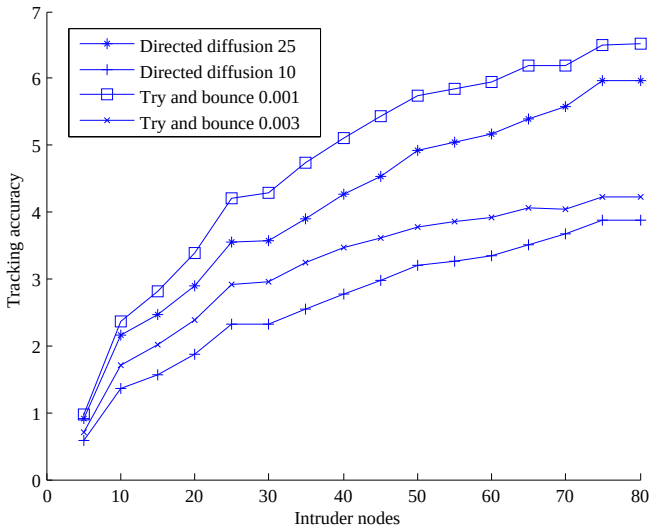

Fig. 3. Average tracking accuracy function of number of intruders.

scenario. The stealth value was measured at the end of the scenario.

Figure 2 shows the average stealth values. The first observation is that, as expected, the stealth values at the end of the scenario decrease with the number of intruder nodes tracked during the scenario. Regardless of the chosen protocol, the more intruders need to be reported, the more stealth is lost.

Again, as expected, for directed diffusion, the longer the reporting interval, the higher the stealth. For TAB, the lower the stealth loss cap, the higher the stealth.

Comparing the two protocols, both TAB instances score significantly higher than either of the directed diffusion instances.

Figure 3 shows the average tracking error for the same experiment runs. Overall, the tracking error (which is summed over the targets and time) naturally increases with the number of targets. Comparing the tracking error of the different dissemination protocols, we find DD-10 be the most accurate, followed closely by TAB-0.003 (but, if we look at Figure 2, with a much higher stealth). Next, at some distance, comes DD-25, matched closely by TAB-0.001, but again, TAB-0.001 has a significantly higher stealth level.

Our conclusion from this set of experiments is that virtually any stealth and tracking error level can be achieved by setting the interval parameter of directed diffusion or the stealth loss cap in TAB, but for the same tracking accuracy, TAB will have a significantly higher stealth level. Alternatively, for the same chosen stealth level, TAB will have a lower tracking error.

D. Results: temporal evolution

While the average values give us a good view of the overall performance, let us now consider the temporal evolution of the measured performance parameters. We choose the scenario with 40 intruder nodes.

Figure 4 show the evolution in time of the stealth value over the $[0 \ldots 5000]$ time interval. All four protocol instances show the expected, monotonically decreasing tendency until the final

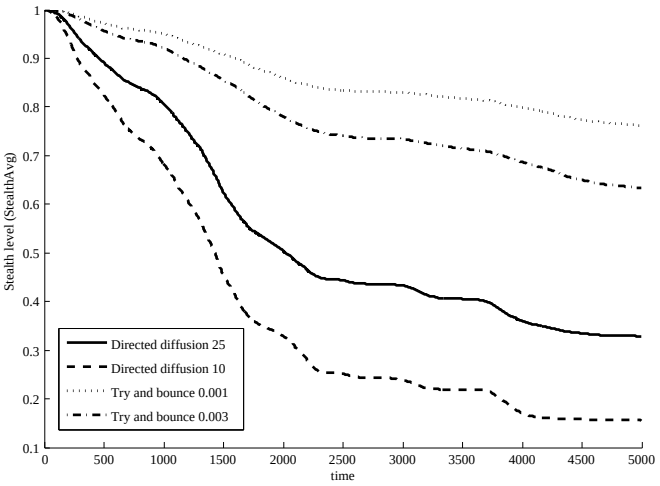

Fig. 4. Time series of the evolution of the stealth level.

values are reached. The graph confirms the fact that the stealth is changing relatively smoothly, but not uniformly over time. The stealth might decrease more or less quickly depending on the dissemination protocol, the number of intruder nodes operating in the field as well as the difficulty of the situations created on a moment by moment basis by the scenarios. We already seen that the try and bounce protocol instances TAB-0.001 and TAB-0.003 are decreasing in stealth much slower than the directed diffusion instances DD-10 and DD-30. An additional observation is that the try and bounce graphs are smoother than the directed diffusion graphs. The reason for this is that for try and bounce, the stealth expenditure is specified directly through the stealth loss cap, while for directed diffusion this is specified only indirectly.

Figure 5 shows the tracking error in function of time. This graph covers a much shorter time scale of only 100 seconds (as the tracking error varies extensively in time, graphs on longer timescales are too cluttered for reading). For all protocol

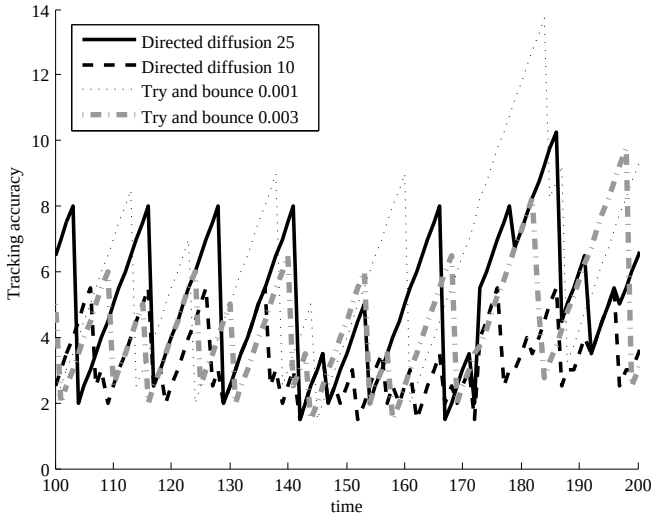

Fig. 5. Tracking accuracy time series.

instances, the tracking error evolves in a sawtooth pattern: the error decreases in a sudden jump whenever a report arrives and then increases in the absence of reports as the intruder node is moving away from the reported position.

The accuracy over larger time frames is affect by several factors. Protocol instances which report more often (such as DD-10 or TAB-0.001) obtain a higher accuracy because the estimate does not have time to drift too far from the correct location. Another factor is the accuracy of the report when received, which in our case it is determined by the delay in the forwarding of the report. Everything else being the same, TAB reports might travel a longer path than directed diffusion ones, due to the occasional bouncing and traveling on paths avoiding the intruder nodes. This is reflected by the fact that the bottom of the sawtooth pattern occasionally stops higher for TAB, which indicates that the report did not arrive on the shortest path. However, this relatively minor difference is more than compensated by the increase in stealth.

VI. Conclusions

In this paper we described an approach to quantify the stealthiness of a node in a sensor network and proposed a dissemination algorithm, TAB which takes into consideration stealthiness when making transmission decisions. We have shown that TAB obtains significantly better stealth level than directed diffusion for equivalent tracking accuracy.

As the issue of stealth has been at best marginally addressed in the current sensor network literature, the scope for future work is very wide. First, the current TAB algorithm can be probably improved by adding more complex inferences - although these will need to be balanced against the increased energy consumption. The scenario we considered is only one point in the very large design space of intruder tracking sensor networks. Depending on the scarcest resource of the current setup, algorithms trying to improve stealth are facing different tradeoffs. Finally, in certain networks, for instance those involving mobile nodes, the wireless transmissions are only one of the many stealthiness risk factors. The reduction of wireless transmissions needs to be balanced against factors such as visual exposure, stealthy or exposed locations, or disclosure due to the sensing actions, such as the use of radar.

References

[1] S. Kumar, T. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *MobiCom '05: Proceedings of the 11th Annual Int. Conf. on Mobile Computing and Networking*, August - September 2005, pp. 284–298.

[2] D. Cook, P. Gmytrasiewicz, and L. Holder, "Decision-theoretic cooperative sensor planning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1013–1023, Oct 1996.

[3] A. Czarlinska and D. Kundur, "Event-driven visual sensor networks: Issues in reliability," in *WACV '08: Proceedings of IEEE Workshop on Applications of Computer Vision*, January 2008, pp. 1–6.

[4] K.-S. Hung, K.-S. Lui, and Y.-K. Kwok, "A trust-based geographical routing scheme in sensor networks," in *WCNC '07: Proceedings of IEEE Wireless Communications and Networking Conference*, March 2007, pp. 3123–3127.

[5] S. Al-Wakeel and S. Al-Swailemm, "Prsa: A path redundancy based security algorithm for wireless sensor networks," in *WCNC '07: Proceedings of IEEE Wireless Communications and Networking Conference*, March 2007, pp. 4156–4160.

[6] D. An and H. Cam, "Route recovery with one-hop broadcast to bypass compromised nodes in wireless sensor networks," in *WCNC '07: Proceedings of IEEE Wireless Communications and Networking Conference*, March 2007, pp. 2495–2500.

[7] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Node compromise modeling and its applications in sensor networks," in *ISCC '07: Proceedings of IEEE Symposium on Computers and Communications*, July 2007, pp. 575–582.

[8] J. Deng, R. Han, and S. Mishra, "INSENS: Intrusion-tolerant routing for wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 216 – 230, January 2006.

[9] B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, and V. Lesser, "Distributed sensor network for real time tracking," in *AGENTS '01: Proceedings of the 5th Int. Conf. on Autonomous Agents*, May - June 2001, pp. 417–424.

[10] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden markov model sensors," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1382–1397, Jun 2002.

[11] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *MobiCom '04: Proceedings of the 10th Annual Int. Conf. on Mobile Computing and Networking*, September - October 2004, pp. 129–143.

[12] T. Yan, T. He, and J. Stankovic, "Differentiated surveillance for sensor networks," in *SenSys '03: Proceedings of the 1st Int. Conf. on Embedded Networked Sensor Systems*, November 2003, pp. 51–62.

[13] S. Olariu, M. Eltoweissy, and M. Younis, "ANSWER: AutoNomouS netWorked sEnsoR system," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 111–124, January 2007.

[14] Y. Wang, X. Wang, B. Xie, D. Wang, and D. Agrawal, "Intrusion detection in homogeneous and heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 698–711, June 2008.

[15] Y. Zou and K. Chakrabarty, "Distributed mobility management for target tracking in mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 872–887, August 2007.

[16] H. Kung and D. Vlah, "Efficient location tracking using sensor networks," in *WCNC '03: Proceedings of IEEE Wireless Communications and Networking Conference*, March 2003, pp. 1954–1961.

[17] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks." in *MOBICOM '00: Proceedings of the 6th Annual Int. Conf. on Mobile Computing and Networking*, August 2000, pp. 56–67.

[18] L. Bölöni and D. Turgut, "YAES - a modular simulator for mobile networks," in *MSWiM '05: Proceedings of the 8th ACM/IEEE Intl. Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, October 2005, pp. 169–173.