

Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks

Wei-Peng Chen, Jennifer C. Hou and Lui Sha

Department of Computer Science

University of Illinois at Urbana-Champaign, Urbana, IL 61801

{wchen3,jhou,lrs}@cs.uiuc.edu

Abstract—In the paper, we devise and evaluate a fully decentralized, light-weight, dynamic clustering algorithm for target tracking. Instead of assuming the same role for all the sensors, we envision a hierarchical sensor network that is composed of (a) a static backbone of sparsely placed high-capability sensors which will assume the role of a cluster head (CH) upon triggered by certain signal events; and (b) moderately to densely populated low-end sensors whose function is to provide sensor information to CHs upon request. A cluster is formed and a CH becomes active, when the acoustic signal strength detected by the CH exceeds a pre-determined threshold. The active CH then broadcasts an information solicitation packet, asking sensors in its vicinity to join the cluster and provide their sensing information.

We address and devise solution approaches (with the use of Voronoi diagram) to realize dynamic clustering: (I1) how CHs cooperate with one another to ensure that for the most of time only one CH (preferably the CH that is closest to the target) is active; (I2) when the active CH solicits for sensor information, instead of having all the sensors in its vicinity reply, only a sufficient number of sensors respond with non-redundant, essential information to determine the target location; and (I3) both packets with which sensors respond to their CHs and packets that CHs report to subscribers do not incur significant collision. Through both probabilistic analysis and ns-2 simulation, we show with the use of Voronoi diagram, the CH that is usually closest to the target is (implicitly) selected as the leader and that the proposed dynamic clustering algorithm effectively eliminates contention among sensors and renders more accurate estimates of target locations as a result of better quality data collected and less collision incurred.

I. INTRODUCTION

With the advancement of MEMS technologies, sensor networks have opened new vistas for a wide range of application domains. These sensor networks usually comprise small, low-power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities. One of their most important applications is target tracking, with the targets to be tracked ranging from security attacks in the forms of chemical, biological, or radiological weapons, to moving objects in civil surveillance, and to changes in light, temperature, pressure, acoustics in environmental monitoring. The type of signals to be sensed are determined based on the types of objects to be tracked.

In spite of the different targets to be tracked and the various signals to be sensed, tracking applications share several com-

mon characteristics: First, the tracking system should report the location of the target to subscribers (usually remote controllers) accurately and in a timely manner. Second, because the data collected by sensors may be redundant, correlated, and/or inconsistent, it is desirable to have sensors collaborate on processing the data and transporting a concise digest to subscribers. This reduces not only the number of packets to be transported, but also the possibility of collision and interference in the shared media. Localized and collaborative data processing also aids in reducing the power consumed in communication activities and hence prolonging the lifetime of sensor networks.

To facilitate collaborative data processing in target tracking-centric sensor networks, the cluster architecture is usually used in which sensors are organized into clusters, with each cluster consisting of a cluster head (CH) and several neighboring sensors (members). In the conventional cluster architecture, clusters are formed statically at the time of network deployment. The attributes of each cluster, such as the size of a cluster, the area it covers, and the members it possesses, are static. In spite of its simplicity, the static cluster architecture suffer from several drawbacks. First, fixed membership is not robust from the perspective of fault tolerance. If a CH dies of power depletion, all the sensors in the cluster render useless. In the case that sensors die, a cluster may not have sufficient sensors to carry out its tracking tasks. Second, fixed membership prevents sensor nodes in different clusters from sharing information and collaborating on data processing. Lastly, fixed membership cannot adapt to highly dynamic scenarios of interest in which sensors in the region of high (low) event concentration may be instrumented to stay awake (go to sleep).

Dynamic cluster architectures, on the other hand, offer several desirable features. Formation of a cluster is triggered by certain events (e.g., detection of an approaching target with acoustic sounds). Specifically, when a sensor with sufficient battery and computational power, detects with a high signal-to-noise ratio (SNR), certain signals of interest, it volunteers to act as a CH. No explicit leader (CH) election is required, and hence no excessive message exchanges are incurred. On the other hand, a judicious, decentralized approach has to be used to ensure that for most of the time only one CH is active in the vicinity of a target to be tracked. Sensors in the vicinity of the active CH are

This work is supported in part by the MURI program N00014-01-0576, by the DARPA NEST program F33615-5-01-c-1905, and by the NSF ANI under grant ANI-0221357.

“invited” to become members of the cluster and will report their sensor data to the CH. In this manner, a cluster is only formed in the area of high event concentration. Sensors do not statically belong to a cluster, and may support different clusters at different times. Moreover, as only one cluster is active in the vicinity of a target, redundant data is suppressed and potential interference and contention at the MAC level is mitigated.

In this paper, we devise and evaluate a fully decentralized, light-weight, dynamic clustering algorithm for single target tracking. We focus on acoustic target tracking, although the proposed approaches can be readily applied to other types of tracking applications. In acoustic tracking, two most common methods for target localization are the time delay-based [6] and energy-based [4], [5] approaches. While time delay-based approaches are susceptible to estimation errors in time synchronization, on-set detection and echo effect, energy-based approaches are more robust in these aspects. Hence we adopt in this paper the energy-based approach as the localization method. Sensors in the acoustic tracking systems perform two types of computation: (1) sensing the energy level of signals; (2) sound analysis, classification, and data fusion. The former is not computational intensive and can be handled by sensors with minimal computation power. The later, however, requires much higher computation power. To this end, we envision a hierarchical sensor network that is composed of (a) a static backbone of sparsely placed high-capability sensors which will assume the role of a CH upon triggered by certain events of interest; and (b) moderately to densely populated low-end sensors whose function is to provide sensor information to CHs upon request. A cluster is formed and a CH becomes active in an on-demand fashion, when the acoustic signal strength detected by the CH exceeds a pre-determined threshold. The active CH then broadcasts an information solicitation packet, asking sensors in its vicinity to join the cluster and provide their sensing information. After receiving a sufficient number of replies from sensors, the CH applies a localization method to estimate the location of the target and send a report to the subscribers.

There are several issues that we have to address and devise solution approaches in order to realize the notion of dynamic clustering: (I1) how CHs cooperate with one another to ensure that for most of the time only one CH (preferably the CH that is closest to the target¹) is active; (I2) when the active CH solicits for sensor information, instead of having all the sensors in its vicinity reply, only a sufficient number of sensors respond with non-redundant information to determine the target location; and (I3) both packets with which sensors respond to their CHs and packets that CHs report to subscribers do not incur significant collision.

¹The reason why the CH closer to the target should be activated is because the quality of sensing data decreases with the distance from the target.

To deal with these issues, we propose, with the use of Voronoi diagram, a probabilistic leader volunteering procedure and a sensor replying method. Initially, we enable all the sensors to calibrate their relative positions to their neighbors (at the CH ↔ CH level and the sensor ↔ sensor level) at the time of network deployment. Then, with the use of Voronoi diagram, each CH (or sensor) can calculate and tabulate the probability that given an distance estimate between a target and itself, the CH (sensor) is closest to the target. This information is used to set up the timer used by a CH to announce its willingness to be active in the leader volunteering process. If no other CHs volunteer before the timer expires, the CH becomes active; otherwise, it suppresses its timer. Similarly, this probabilistic information is also used by a sensor to determine whether or not it should respond to a CH upon request to provide sensor information and the timer value with which it responds to such a request. Note that due to the limited mobility nature of sensors, the calibration and tabulation process needs only to be carried out initially at network deployment and infrequently during system operation.

Research in using sensor networks for target tracking has recently attracted much attention. The type of work can be roughly categorized into two categories. In the first category, data captured in different sensors are collected at a node, and signal processing schemes, such as Maximum Likelihood testing (ML) [5] and minimum square estimation [4], are applied to localize the target. In the second category, the target location is estimated successively based on the current measurement at a sensor and the past history at other sensors [8], [2], [3]. These types of work mainly focus in the signal processing aspect, rather than the communication aspect, of target tracking. In contrast, our proposed approach balances in selecting a subset of most adequate sensors to conduct tracking tasks. The signal processing techniques mentioned above can be incorporated with the proposed approach in a complementary manner. Clustering techniques, on the other hand, have been proposed in wireless networks for routing [14], [13], [7], bandwidth reuse [11], and data gathering [12]. None of them are targeted for the purpose of target tracking and hence do not consider issues (I1 – I3) comprehensively.

Through both the probabilistic analysis and *ns-2* simulation, we evaluate and demonstrate the effectiveness of the proposed approaches. In particular, we show via a simplified model (which captures all the essential properties) that the probability that packets collide with one another is very small under the proposed approaches. We also show via simulation that with the use of Voronoi diagram, the CH that is usually closest to the target is (implicitly) selected as the leader and that the proposed dynamic clustering algorithm effectively eliminates contention among sensors and renders better and more accurate estimates on target location as a result of better quality data collected and

less collision incurred. As compared to the performance of the static cluster, the proposed approaches reduce the estimation error and latency by 23% and 15%, respectively.

The rest of the paper is organized as follows. We lay the necessary technical background and give an overview of the proposed dynamic clustering algorithm in Section 2. We then delve into the algorithm details in Section 3. Following that we present, respectively, the analysis and simulation results in Sections 4–5. Finally we conclude the paper in Section 6.

II. SYSTEM OVERVIEW

A. Energy-Based Localization

The fundamental principle applied in the energy-based approach [4], [5] is that the signal strength (i.e., energy) of a received signal decreases exponentially with the propagation distance [2], [4]:

$$r_i = a \cdot \|x - x_i\|^{-\alpha} + w_i, \quad 1 \leq i \leq N, \quad (1)$$

where r_i is the received signal strength in the i^{th} sensor, $a \in \mathcal{R}$ is the (unknown) strength of an acoustic signal from the target, $x \in \mathcal{R}^2$ is the target position yet to be determined, $x_i \in \mathcal{R}^2$ is the known position of the i^{th} sensor, α is the (known) attenuation coefficient, and w_i is the white Gaussian noise with zero-mean and variance σ^2 .

Instead of solving for the unknowns in Eq. (1) based on energy readings from multiple sensors, we use the following simple Voronoi diagram-based approach. Conceptually, each pair of energy readings, (r_i, r_j) , from two sensors i and j determines a half plane that contains the target, i.e., if $r_i > r_j$, the target is closer to sensor i than to sensor j and hence lies in the half plane that contains sensor i . With multiple pairs of energy readings, the target location can be confined to be the intersection area of all the half planes. It can be shown that for a set of n sensor readings, $n - 1$ out of the total $n(n - 1)/2$ half planes are independent. Therefore, in order to obtain a bounded intersection area, at least four sensor readings are required. As the locations of sensors are static (only subject to environmental factors such as wind), the intersection area can be determined in advance in the form of Voronoi diagram. For completeness of the paper, we define the Voronoi diagram below [1]:

Definition 1: (Voronoi) Let $P = \{p_1, \dots, p_N\}$ be a finite set of points in the n -dimensional space \mathcal{R}^n and their location vectors $l_i \neq l_j, \forall i \neq j$. The region given by $\mathcal{V}(p_i) = \{l \mid \|l - l_i\| \leq \|l - l_j\|, \forall j \neq i\}$ is called *Voronoi cell* associated with p_i and $\mathcal{V}(P) = \bigcup_{i=1}^N \mathcal{V}(p_i)$ is said to be the *Voronoi diagram* of P .

We claim that the Voronoi cell $\mathcal{V}(p_i)$ associated with the point p_i is exactly the intersection area of all the half planes by the following observation:

Observation 1: $\mathcal{V}(p_i) = \bigcap_{1 \leq j \leq N, j \neq i} h(p_i, p_j)$, where $h(p_i, p_j)$ is the open half plane containing p_i .

As a result, as long as the sensor with the maximum energy reading, say p_i , can be identified, the location of the target is inside $\mathcal{V}(p_i)$ and the position of p_i can be used as the *approximate* location of the target. Note that the error of the above approximation is small and bounded if the target is at a bounded Voronoi cell $\mathcal{V}(p_i)$. In the case that the Voronoi cell identified is not bounded, more detailed estimation is needed. For example, as indicated in [4], each pair of energy readings determines a circle on which the target may reside. To estimate the target location, a nonlinear least square equation has to be solved. Considering the limited computational power sensors are equipped with, we assume in this paper a target travels within a bounded region, and it is sufficient to estimate the location of the target with the position of the sensor with the maximum signal strength.

B. Overview of Proposed Dynamic Clustering Algorithm

As mentioned in Section 1, we envision a heterogeneous, hierarchical sensor network that is composed of (a) a static backbone of sparsely placed high-capability sensors called CHs; and (b) moderately to densely populated low-end sensors whose function is to provide sensing information to CHs upon requests. Because of the limited mobility nature of sensors, the calibration process of sensor locations is executed only once when the network is deployed. In this calibration process, geographical information required to construct the Voronoi diagram is collected, and the Voronoi diagram constructed. Furthermore, both CHs and sensors construct several tables to facilitate determination of the back-off timer values (to be used when a CH intends to volunteer itself as a leader and when a sensor intends to respond to a CH).

A CH volunteers to become active when it detects that the strength of an acoustic signal it receives exceeds a pre-determined threshold and the signal matches one of the signal patterns which the system intends to track. As multiple CHs may detect the acoustic signal with a sufficiently high signal-to-noise ratio and volunteer themselves as active leaders, we devise in Section 3.2 a two-phase volunteering procedure to determine in a decentralized manner the CH with the strongest signal strength.

The tasks of an active CH include the following four steps:

- 1) broadcasting a packet that contains the energy and the extracted signature² of the detected signal to sensors,
- 2) receiving replies from sensors,
- 3) estimating the location of the target based on replies,
- 4) sending the result to subscriber(s).

²The extracted signature can be either the raw data or the extracted feature of a signal.

Upon receipt a broadcast packet from a CH, a sensor matches the signature with its buffered data. In the case of a match, the sensor then determines, with the use of the Voronoi diagram based table, whether or not it may be (i) the sensor that is closest to the target or (ii) one of the neighbors of the sensor that is closest to the target. If any of the above two conditions holds, it replies, after a random delay, to the CH the strength of the signal it receives. The random delay is determined based on the strength of the signal the sensor receives so as to mitigate collision. We will elaborate on how a sensor determines whether or not, and when, it should reply to a broadcast message from a CH in Section 3.3.

Once the CH collects enough replies, it ignores all subsequent replies (if any), generates the localization result and sends the result back to subscribers. Sensors that decided to reply but have not yet done so (as their timers have not expired) stop replying, if they overhear the packet that carries the localization result.

The relationship between the radio transmission range and the acoustic signal detection range determines the size of a cluster. The radio transmission range is controlled by adjusting the transmission power, while the acoustic signal detection range is controlled by adjusting the detection threshold. If the acoustic signal detection range is larger than the radio transmission range, multiple CHs may become active and multiple clusters formed at the same time, without knowing the existence of one another. The results obtained from different clusters may differ dramatically because a CH may not be able to recruit sufficient sensors and gather enough sensor information to confine the target in a bounded Voronoi cell. On the other hand, if the radio transmission range is larger than the acoustic signal detection range, the localization results will be more accurate but collision has to be handled carefully. In this paper, we assume that the radio transmission range is set to be twice of the acoustic signal detection range.

III. DETAILED DESCRIPTION OF PROPOSED DYNAMIC CLUSTERING ALGORITHM

The proposed dynamic clustering algorithm is composed of four component mechanisms: initial distance calibration and tabulation, CH volunteering, sensor replying, and reporting of tracking results. In what follows we elaborate on each of the four component mechanisms.

A. Distance Calibration and Tabulation

Before the acoustic tracing system starts to function, each sensor has to know the positions of sensors in its tracking ranges. Under the assumption that the radio transmission range is larger than the acoustic detection range, a sensor can notify other sensors of its ID, device function (CH or sensor), and location information by broadcast. To reduce the possibility of collisions,

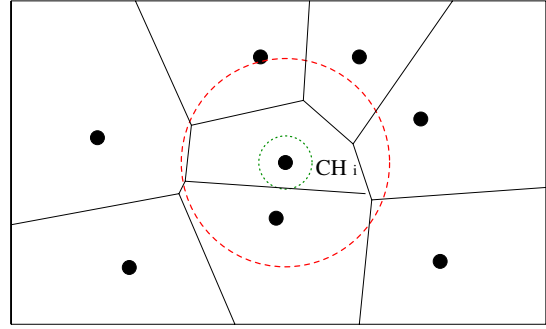


Fig. 1. Voronoi diagram of CH_i . If a target locates within the inner dotted circle, CH_i is sure that the target is in its Voronoi cell. On the other hand, if a target resides outside the outer dashed circle, CH_i is sure that the target is out of its Voronoi cell.

a broadcast packet is delayed by a back-off value determined based on the sensor ID.

Construction of Voronoi diagrams at a CH: After receiving the location information from all the neighboring sensors, a CH constructs two Voronoi diagrams around itself, one for the set of neighboring sensors and the other for the set of neighboring CHs. In what follows, we first elaborate on how to use Voronoi diagrams to construct response tables and will explain their usage in Sections 3.2–3.3.

Construction of the response table based on Voronoi diagrams at a CH: After the Voronoi diagrams are constructed, a CH proceeds to construct the *response table* to facilitate determination of back-off timer values (to be used when the CH intends to volunteer itself as a leader). The table is indexed by the estimated distance from the target to the CH, d , and each table entry stores the conditional probability that with the distance d , a target indeed locates within this CH's Voronoi cell.

The distance from the target to a CH, say CH_i , can be estimated (with the noise ignored and the conjectured signal strength from the target³) as $d = (r/a)^{-1/\alpha}$. That is, the possible position at which the target may be located is a circle centered at CH_i and with radius d . Next, we derive the conditional probability, $\Pr(i|d)$, that the target locates within the Voronoi cell of CH_i , given the distance from the target to CH_i , d . Let $d_{i,min}$ and $d_{i,max}$ denote, respectively, the distance from CH_i to its nearest neighboring CH and that from CH_i to the farthest Voronoi vertex of its Voronoi cell. We have to consider three cases (Figure. 1)

- 1) $d < 1/2 \cdot d_{i,min}$: CH_i is the nearest CH to the target because the circle that is centered at CH_i and has a radius of d lies completely within CH_i 's Voronoi cell. Hence $\Pr(i|d) = 1.0$.
- 2) $d > d_{i,max}$: By the definition of $d_{i,max}$, the circle cen-

³Initially a is set to a default value and dynamically adjusted according to the previous localization results.

```

CALCPr(i|d)(d)
1. gain ← 0; loss ← 0
2. for j ← 1 to resolution
3.   θ ← 2π · j/resolution
4.   x ← CHi.x + d · cos(θ); y ← CHi.y + d · sin(θ)
5.   if (x, y) is within Voronoi cell V(CHi)
6.     gain++
7.   else if (x, y) is within V(CHk) and dist((x, y), CHk)
8.     > 1/2 · dk,min
9.     loss++
9. return gain/(gain+loss)

```

Fig. 2. The algorithm that calculates $\Pr(i|d)$ when $1/2 \cdot d_{k,min} < d < d_{i,max}$.

tered at CH_i with a radius of d lies completely outside CH_i 's Voronoi cell. Hence $\Pr(i|d) = 0.0$.

- 3) $1/2 \cdot d_{i,min} < d < d_{i,max}$: The circle that is centered at CH_i and has a radius of d is partially located within CH_i 's Voronoi cell and hence $0.0 \leq \Pr(i|d) \leq 1.0$. We will further estimate $\Pr(i|d)$ using the algorithm given in Fig. 2.

Note that lines 5 and 7 in the algorithm can be executed in $O(\log n)$ time and $O(n)$ space, where n is the number of CHs in the detection range, after the Voronoi diagram is constructed. Essentially the algorithm takes $(360/resolution)$ samples on the circle that is centered at CH_i and has a radius of d , where *resolution* is a tunable parameter. Whenever a sample lies within $V(CH_i)$, the variable *gain* is incremented. On the other hand, when a sample may lie within $V(CH_k)$ for some neighboring CH_k , the variable *loss* is incremented, except that samples that are surely located in the Voronoi cell of one neighboring CH are not counted in *loss*. The probability sought for $\Pr(i|d)$ is then estimated as $\frac{gain}{gain+loss}$.

Note also that in line 7 CH_i needs to know $d_{k,min}$ for every neighboring CH_k . This can be achieved by having each CH broadcast the distance to its nearest CH in the second round. Also, if CH_i 's Voronoi cell is not bounded, the algorithm excludes in line 5 samples which are in $V(CH_i)$ but with the distance to CH_i larger than that from CH_i to its farthest Voronoi vertex.

To keep the table size at a fixed value, we quantize the distance from the target to itself d as follows. The table contains k entries. The first entry is indexed by the maximum value of d (denoted as d_{min}) such that $\Pr(i|d_{min}) = 1.0$, while the last entry by the minimum value of d (denoted as d_{max}) such that $\Pr(i|d_{max}) = 0.0$. The other $k - 2$ entries are indexed by values evenly spaced between d_{min} and d_{max} . Given an arbitrary distance d , a binary search is made to locate the two entries with the closest distance and interpolation is used to obtain the approximate value of $\Pr(i|d)$.

Construction of the Voronoi diagram and the response tables at a sensor: A sensor S_j constructs one Voronoi diagram

around itself for the set of neighboring sensors. The response table is indexed by the ratio, $r_{i \rightarrow j}$, of the signal strength at a CH_i to its received energy, where CH_i is the CH that initiates the solicitation. Each table entry stores the conditional probability, $\Pr(j|r_{i \rightarrow j})$, that the target locates in S_j 's Voronoi cell $V(S_j)$ rather than other neighboring sensors or CHs, given the ratio $r_{i \rightarrow j}$. Specifically, suppose a sensor S_j , located at (x_2, y_2) , receives an information solicitation packet from a CH CH_i , located at (x_1, y_1) . Assume that the signal strength detected at CH_i and S_j , is, respectively, r_1 and r_2 . Then $r_{i \rightarrow j} = \frac{r_1}{r_2} = (\frac{d_1}{d_2})^{-\alpha}$. Let $c \triangleq (\frac{d_1}{d_2})^2 = (\frac{r_1}{r_2})^{-2/\alpha}$. After a few algebraic operations, one can derive that the locus of the potential position of the target, (x, y) , is a circle characterized by

$$(x - x_1)^2 + (y - y_1)^2 = c(x - x_2)^2 + c(y - y_2)^2$$

or

$$(x - o_x)^2 + (y - o_y)^2 = \rho^2,$$

where $(o_x, o_y) = (\frac{cx_2 - x_1}{c-1}, \frac{cy_2 - y_1}{c-1})$ is the center of the circle and $\rho = \sqrt{\frac{c}{(c-1)^2} \cdot ((x_1 - x_2)^2 + (y_1 - y_2)^2)}$ is the radius of the circle.

Given the above expressions and the signal strength detected at CH_i , each sensor can locate the potential positions of the target. Using a similar algorithm to that in Fig. 2, we can calculate and tabulate $\Pr(j|r_{i \rightarrow j})$. Each sensor S_j then maintains, for each CH within its transmission range, a table of k entries. In addition, let \overline{N}_j denotes the event that the target is in neither $V(S_j)$ nor the Voronoi cells of any of S_j 's Voronoi neighbors. Then each sensor calculates the conditional probability, $\Pr(\overline{N}_j|r_{i \rightarrow j})$, that the target is located at neither $V(S_j)$ nor the Voronoi cells of any of S_j 's Voronoi neighbors. Note that S_j needs only to store the minimum value of $r_{i \rightarrow j}$, r_{min} , such that $\Pr(\overline{N}_j|r_{min}) = 1.0$.

One point is worthy of mentioning — the process of table construction takes place only once at each CH/sensor. In the case that sensors are not equipped with sufficient circuitry to perform this pre-processing, their nearest CH may construct the tables on their behalves and transmit the resulting tables to them.

B. Cluster Head Volunteering

In the first step of dynamic clustering, a CH volunteers to recruit sensors to form a cluster if its detected signal strength exceeds a predefined threshold. One fundamental design issue to consider is which CH(s) should be elected to form a cluster if more than one CH detect simultaneously signals with the strength exceeding the threshold. Note that if two or more clusters are active simultaneously, packets exchanged in one cluster may interfere/collide with those in the other cluster(s). (This is corroborated by our performance study in Sec. 4 in which we

will investigate the performance in the case that more than one cluster is active.) Ideally, the CH that is closest to the target (or the CH with the largest SNR ratio) should be elected. Since the communication cost of deterministic leader election is very high, we propose to use a two-phase, random delay-based broadcast mechanism to implicitly determine the active CH. In what follows, we first describe how the random delay is set and then delve into the two-phase broadcast mechanism.

Determination of back-off timer values: Without a centralized facility and/or excessive message exchanges for CH election, the most effective method to determine the active CH is to figure in the received signal strength into the determination of the back-off timer values used to send solicitation packets. A CH whose received signal strength exceeds the pre-determined threshold sets a back-off timer and does not broadcast its solicitation packet until the timer expires. If by the time the back-off timer expires, the CH receives a solicitation packet from some other CH, it cancels the timer. Specifically, the back-off time, D , for which CH_i (with the estimated distance d) delays before sending its solicitation packet is:

$$D = W_{min} + (W_{max} - W_{min}) \cdot (1 - \Pr \{ \xi | d \}) + U(W_{ran}), \quad (2)$$

where W_{min} and W_{max} denote the minimum and maximum backoff timer values, $U()$ is the uniform distribution in $[0, W_{ran} - 1]$, and $\Pr \{ \xi | d \}$ can be retrieved from the response table (Section 3.1). Note that D contains two parts. The first two terms in Eq. (2) are the deterministic part that relates the estimated distance to the back-off delay value, and the third term accounts for the random part that prevents potential collision when the distances from the target to two or more CHs are approximately the same. The random part is an order of magnitude smaller than the deterministic part. Note also that the back-off timer is set at the application level, meaning that a CH only passes an information solicitation broadcast packet onto its MAC layer when the back-off timer expires. An underlying carrier-sense MAC protocol, such as a light-weight version of IEEE 802.11b, is still needed to mitigate collision at the MAC level.

Two-phase broadcast with energy and signature packets: Although the above random back-off delay method reduces the possibility of collision, it does not totally eliminate it. This is in part due to the fact that these information solicitation packets may be queued at the MAC layer before being transmitted. Moreover, in order for sensors to identify which event a CH is interested in, information solicitation packets contain not only the signal strength but also the signature of the acoustic sound and as a result are usually quite large. A sensor then attempts to match the received signature with its buffered data to make sure the same event is being tracked. These solicitation packets of large sizes also contribute to the possibility of collision.

To deal with the above problems, we propose a two-phase broadcast mechanism: in the first phase, an *energy packet* that carries only the signal strength information is broadcast. In the second phase, a *signature packet* that contains the detailed signature information is then broadcast. Both packets are subject to the same random back-off delay value (except for the randomized term). Specifically the two-phase broadcast mechanism operates as follows:

- (R1) A CH sets its back-off timer with the value of D (Eq. (2)) for the energy packet.
- (R2) When the timer in the first phase expires, the CH broadcasts an energy packet and sets its back-off timer to the same value in the second phase (except for the randomized term).
- (R3) When the timer in the second phase expires, the CH broadcasts a signature packet.
- (R4) If by the time the timer expires in the first or second phase, the CH overhears (i) a broadcast packet with the signal strength larger than that it itself detects or (ii) a signature packet, it cancels its timer and does not henceforth volunteer. Otherwise, the overheard packet is ignored.

Note that the energy packet in the first phase is much shorter than the signature packet in the second phase and hence the possibility of collision is reduced. Also, as will be analyzed in detail in Section 4, even if the energy packet of a CH collides with other packets, the CH may still be able to broadcast its signature packet in the second phase.

Operations performed by a volunteering CH: In order to realize the above mechanism, clock synchronization is required in two respects. First, a CH has to inform sensors of the time when the event takes place so that sensors can match the signature contained in the packet with its buffered data. Second, when a CH sends the localization result to the sink, it has to inform the sink of the time when the event takes place. To avoid the expensive clock synchronization process, we use relative clock values rather than absolute clock values. A CH records the time it detects the event. Before broadcasting the signature packet, the CH calculates the time lag (which includes the processing time, back-off delay and expected propagation delay), and attaches the time lag in the signature packet. When a sensor receives the signature packet, it can infer the time when the CH detected the event.

After broadcasting the signature packet, a CH sets a timer to wait for replies from neighboring sensors. If a sufficient number of replies are received before the timer expires, the CH cancels the timer and calculates the localization result; otherwise, the result is generated upon the expiry of the timer.

C. Sensor Replying

After a sensor receives a signature packet, the first step is to match the signature with its buffered data. The search range can be confined with the use of the time lag information in the signature packet. If the signature is matched, the signal strength in the buffered data (as well as the ratio of the signal strength detected at the CH to that in the buffered data) is calculated.

A sensor S_j does not respond if $\Pr(\hat{q}|r_{i \rightarrow j}) = 0$ and $r_{i \rightarrow j} \geq c' \cdot r_{min}$, where c' is a constant and r_{min} is the minimum value of $r_{i \rightarrow j}$ such that $\Pr(\bar{N}_j|r_{min}) = 1.0$. In the case that a response is to be sent, a similar, random back-off method is used to avoid potential collision of all the replies from sensors to CH_i . That is, a sensor S_j delays its reply by a back-off value determined by

$$D' = W'_{min} + (W'_{max} - W'_{min}) \cdot (1 - \Pr(j|r_{i \rightarrow j})) + U(W'_{ran}),$$

where W'_{min} and W'_{max} are the minimum and maximum back-off values, and $\Pr(j|r_{i \rightarrow j})$ can be retrieved from the response table (Section 3.1).

If by the time the back-off timer expires the sensor overhears reply packets from other sensors, it records the sensor that reports the largest signal strength. When the timer expires, the sensor sends a reply packet only if (i) the signal strength it detected is larger than that carried in any of the overheard reply packets; or (ii) it is one of the Voronoi neighbors of the sensor node that reports the largest signal strength. The two conditions ensure that the set of replies that the active CH will receive includes a reply from a node S_j with the largest signal strength, and replies from all S_j 's Voronoi neighbors. Due to the convexity of the propagation model, the target is guaranteed to be within the Voronoi cell of S_j , $\mathcal{V}(S_j)$. Finally, when a sensor overhears a tracking report packet from a CH, if it prepares to reply to this CH and the event ID is the same, the sensor cancels its timer and does not respond.

D. Reporting Tracking Results

A CH generates the localization result, if either the timer set after the signature packet was sent expires or the CH receives a sufficient number of replies, whichever occurs first. By "sufficient," we mean the set of replies includes a reply from a node S_j with the largest signal strength, and replies from all S_j 's Voronoi neighbors. In this manner, the target is guaranteed to be within the Voronoi cell of S_j , $\mathcal{V}(S_j)$, due to the convexity of the propagation model. Then, a simple localization approach is used to determine the position of the target: the position of the sensor that reports the largest signal strength is taken as the estimate of the target. A more sophisticated localization method is currently under investigation and will be figured in our future work.

Once the localization result is generated, the CH has to send the location and time information of the target to the sink(s).

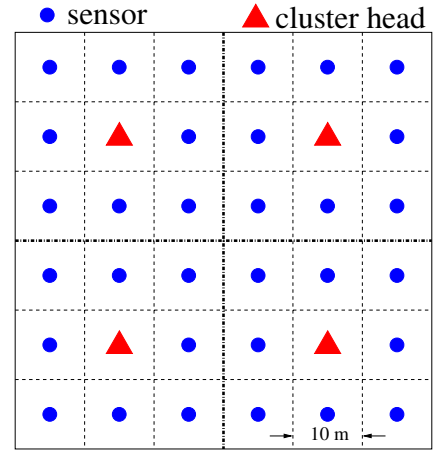


Fig. 3. The scenario that shows how CHs and sensors are deployed in the analysis.

The CH calculates the time difference between the time instant when it detected the event and the expected time instant when the packet arrives at the next hop and sends the tracking report packet. Each intermediate router accumulates and updates the time lag in the same manner. When the sink receives the report, it can calculate the time when the CH detected the event.

IV. ANALYSIS OF PROPOSED ALGORITHM

In this section we analyze the performance of the CH volunteering procedure (Section 3.2) and show that with properly selected parameters W_{min} , W_{max} , and W_{ran} , (a) the CH with the largest signal strength (CH_1) transmits its energy packet earlier than any other CH (and thus suppress other energy packets) with a high probability; and (b) in the case that (a) is not true, CH_1 still has a very good chance to become the leader eventually. To facilitate derivation, we make the following simplifications:

- 1) Both the CHs and the sensors are deployed on a regular square grid shown in Fig. 3. In this scenario, at most four CHs will contend to become a leader.
- 2) The following equation, rather than Eq. (2), is used by CH_i to determine its back-off time:

$$D = W_{min} + (W_{max} - W_{min}) \cdot \left(\frac{d - t_1}{t_2 - t_1} \right) + U(W_{ran}) \quad (3)$$

where d is the estimated distance from the target to CH_i , t_1 is half of the distance from CH_i to the closest neighboring CH, and t_2 is the distance from CH_i to its farthest Voronoi vertex. In the scenario given in Fig. 3, $t_2 = \sqrt{2} \cdot t_1$. Note that we replace $1 - \Pr(\hat{q}|d)$ in Eq. (2) with $\frac{d-t_1}{t_2-t_1}$ to simplify the analysis.

Given the position of a target, we number CHs in the increasing order of their distances to the target, such that if d_i denotes the distance from CH_i to the target, we have $d_1 < d_2 < \dots <$

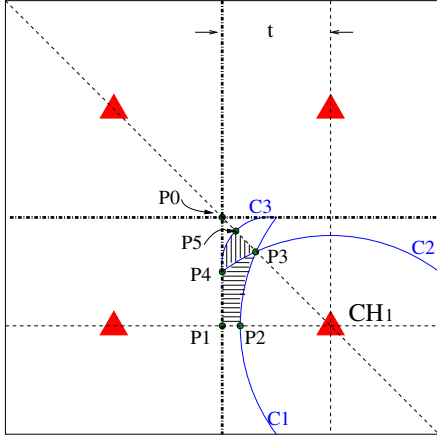


Fig. 4. Regions of possible collisions from CH_1 . $\overline{P_1P_2} = \delta/2$.

d_N . Ideally by the end of the volunteering process, CH_1 should become the leader. Now we consider three cases:

- 1) Case (i): CH_1 transmits its energy packet earlier than any other CHs and by overhearing CH_1 's energy packet, other CHs cancel their back-off timers and do not volunteer.
- 2) Case (ii): CH_i , for some $i > 1$ successfully transmits its energy packet earlier than CH_1 .⁴ CH_1 proceeds to transmit its energy packet by (R4) in Section 3.2, and assume this packet does not collide with CH_i 's signature packet or energy packets from other CHs. In this case, if CH_1 's energy packet is sent earlier than CH_i 's signature packet, CH_1 will still become the leader.
- 3) Case (iii): The energy packet from CH_1 collides with the energy packet from CH_j , $j > 1$. but the energy packet from CH_i , $i > 1$ is successfully transmitted. In this case, CH_1 may still have chance to become active if its signature packet is sent earlier than any other signature packets and does not collide with other packets.

First we show in the following lemma that cases (ii) and (iii) occur with small probabilities, if the parameters W_{min} , W_{max} , and W_{ran} are properly selected.

Lemma 1: Cases (ii) and (iii) occur only if $d_i - d_1 \leq \delta$, where $\delta = \frac{W_{ran}-1}{W_{max}-W_{min}} \cdot (t_2 - t_1)$, where W_{min} and W_{max} are the maximum and minimum backoff timer values and $[0, W_{ran} - 1]$ is the range of the uniform distribution in Eq. (3).

Proof: Cases (ii) and (iii) occur only if $D_i \leq D_1$, for some i , where D_i is the backoff time determined by Eq. (3). That is,

$$D_i = W_{min} + (W_{max} - W_{min}) \cdot \left(\frac{d_i - t_1}{t_2 - t_1} \right) + U(W_{ran})$$

$$\leq W_{min} + (W_{max} - W_{min}) \cdot \left(\frac{d_1 - t_1}{t_2 - t_1} \right) + U(W_{ran}) = D_1.$$

⁴There may be multiple such CHs.

As $U(W_{ran})$ is a discrete random variable with the uniform distribution in $[0, W_{ran} - 1]$, we set $U(W_{ran})$ in D_i to be zero and $U(W_{ran})$ in D_1 to be $W_{ran} - 1$ for the worse case, and can reach the following conclusion

$$d_i - d_1 \leq \frac{W_{ran} - 1}{W_{max} - W_{min}} \cdot (t_2 - t_1) \triangleq \delta.$$

In this paper we set $W_{ran} \ll W_{max}$ (approximately three orders of magnitude smaller) to make δ a very small value, thus ensuring CH_1 transmits its energy packet earlier than any other packets with a high probability. □

Lemma 2: In case (ii), CH_1 will become the leader if $W_{ran} < W_{min} + (W_{max} - W_{min}) \cdot \left(\frac{2d_i - d_1 - t_1}{t_2 - t_1} \right) + 1$.

Proof: As discussed in Section 3.2, after CH_i sends its energy packet, it has to wait for D_i' units of time before sending the signature packet. Moreover, the timer is cancelled and the signature packet is not sent, if CH_i overhears an energy packet with a larger signal strength. Therefore, as long as CH_1 sends its energy packet before $D_i + D_i'$, CH_i will not send the signature packet. That is,

$$W_{ran} < W_{min} + (W_{max} - W_{min}) \cdot \left(\frac{2d_i - d_1 - t_1}{t_2 - t_1} \right) + 1.$$

In this paper we set W_{ran} to be equal to W_{min} , thus ensuring in case (ii) CH_1 always becomes the leader. □

Now we inspect the probability that CH_1 will not become a leader in case (iii). In this case, CH_1 may still become the leader if (a) CH_i 's signature packet is not scheduled before CH_1 's signature packet and (b) CH_1 's signature packet does not collide with any other signature or energy packets. As the probability of the above events can not be exactly and easily calculated, we use the probability that CH_1 's energy packet collides with other energy packets as the upper bound of the probability that CH_1 can not become the leader. This is because if the energy packet from CH_1 does not collide with other packets, CH_1 always become the leader. We will show that this probability is very small.

The first step to calculate the probability that CH_1 's energy packet collides with other energy packets is to identify the region in the Voronoi diagram in which collisions may potentially occur. By Lemma 1, any point in such a region possesses the following property: The distance between the distance from this point to CH_i , d_i , and that to CH_1 , d_1 , is less than δ . As $d_i - d_1 = \delta$ is a hyperbola in the Voronoi diagram, the region is bounded by the hyperbolas and the Voronoi cell boundaries. For example, if the CHs and sensors are deployed in the square grid in Fig. 3, then the hyperbolas that bound the collision region are given in Fig. 4. Because CH_1 's Voronoi cell is symmetric, we only consider its one eighth portion in the northwestern corner (i.e., the area bounded by CH_1 , P_0 , and P_1).

Region	$Pr[CH_1 \text{ collision} p = (x, y)]$
\mathcal{R}_2	$\sum_{t=W_2}^{W_1+W_{ran}-1} \frac{1}{W_{ran}^2} = \frac{W_{ran}+W_1-W_2}{W_{ran}^2}$
\mathcal{R}_3	$\sum_{t=W_2}^{W_3-1} \frac{1}{W_{ran}^2} + \sum_{t=W_3}^{W_1+W_{ran}-1} \left\{ \frac{1}{W_{ran}^3} + \binom{2}{1} \frac{1}{W_{ran}^2} \right\} = \frac{W_3-W_2}{W_{ran}^2} + \left(\frac{1+2W_{ran}}{W_{ran}^3} \right) (W_{ran} + W_1 - W_3)$
\mathcal{R}_4	$\sum_{t=W_2}^{W_3-1} \frac{1}{W_{ran}^2} + \sum_{t=W_3}^{W_4-1} \left\{ \frac{1}{W_{ran}^3} + \binom{2}{1} \frac{1}{W_{ran}^2} \right\} + \sum_{t=W_4}^{W_1+W_{ran}-1} \left\{ \frac{1}{W_{ran}^4} + \binom{3}{2} \frac{1}{W_{ran}^3} + \binom{3}{1} \frac{1}{W_{ran}^2} \right\} = \frac{W_3-W_2}{W_{ran}^2} + \frac{1+2W_{ran}}{W_{ran}^3} (W_4 - W_3) + \frac{1+3W_{ran}+3W_{ran}^2}{W_{ran}^4} (W_{ran} + W_1 - W_4)$

TABLE I

CONDITIONAL PROBABILITY, $Pr(\text{COLLISION OF } CH_1 \text{'S ENERGY PACKET}|p = (x, y))$, THAT CH_1 'S ENERGY PACKET COLLIDES WITH OTHER ENERGY PACKETS WHEN THE TARGET IS LOCATED AT \mathcal{R}_i . W_i IS THE DETERMINISTIC PART IN THE BACKOFF TIMER VALUE OF CH_i .

The collision region under consideration is composed of four sub-regions, $\mathcal{R}_1(CH_1, P_2, P_3)$, $\mathcal{R}_2(P_1, P_4, P_3, P_2)$, $\mathcal{R}_3(P_3, P_4, P_5)$, and $\mathcal{R}_4(P_4, P_0, P_5)$. (Points in the parenthesis represent the vertices of the sub-regions in the clockwise direction.) If the target is located in region \mathcal{R}_1 , by Lemma 1 CH_1 sends its energy packet earlier than any other CHs. On the other hand, if the target is located in region $\mathcal{R}_i, i = 2, 3, 4$, CH_1 's energy packet may collide with $(i - 1)$ other energy packets. For example, if the target is located in region \mathcal{R}_3 , CH_1 's energy packet may potentially collide with energy packets sent by the CHs in the left lower grid and in the right upper grid. Given that the target is located at $p = (x, y)$ in region $\mathcal{R}_i, i = 2, 3, 4$, the conditional probability, $Pr(\text{collision of } CH_1 \text{'s energy packet}|p = (x, y))$, that CH_1 's energy packet collides with other energy packets is given in Table 1. Finally the probability that CH_1 's energy packet collides with other energy packets can be expressed as:

$$Pr(\text{collision of } CH_1 \text{'s energy pkt}) = \frac{2}{t^2} \cdot \sum_{i=2}^4 \left\{ \int_{\mathcal{R}_i} Pr(\text{collision of } CH_1 \text{'s energy pkt}|p = (x, y)) \cdot dx dy \right\}.$$

In this paper we set W_{max} , W_{min} , W_{ran} to be 0.1 second, 10^{-4} second, and 10^{-4} second, respectively.⁵ In this case $\delta = 3.317 \times 10^{-4}t$, and region \mathcal{R}_i is much smaller than region \mathcal{R}_1 . The numerical result of $Pr(\text{collision of } CH_1 \text{'s energy pkt})$ is equal to 5.3×10^{-5} . This ensures the probability that CH_1 's energy packet collides with other energy packets is almost zero, and the proposed algorithm always elects the CH with the largest signal strength to be the leader.

V. SIMULATION RESULTS

In this section, we evaluate the proposed dynamic clustering algorithm using *SensorSim* [10] (which is built upon *ns-2*[9]). The surveillance area is $180 \times 180 m^2$. A total of 324 devices, including 36 CHs and 288 sensors, are deployed in the area. In addition, a sink is located at (0,0) in the system. Two deployment

configurations are considered: one is to deploy the CHs and sensors on a square grid shown in Fig. 3, and the other is to assume both CHs and sensors are uniformly distributed in the area. The detection threshold of each acoustic microphone is adjusted so that the acoustic detection range is 25 meters. The radio transmission range is set to be twice of the acoustic detecting range, i.e., 50 meters. A static topology based sink tree rooted at the sink is built by the underlying routing protocol. IEEE 802.11b is adopted as the underlying MAC protocol.

The performance metrics of interest are (i) the location error: the deviation (in meters) of the estimated location from the exact location of the target; (ii) the latency: the time interval from the instant when the acoustic event occurs till the time instant when the location result is delivered to the sink; (iii) the number of events detected and reported to the sink; (iv) the number of collisions throughout the simulation; and (v) the total number of control messages (information solicitation messages, replies, and tracking reports to the sink) throughout the simulation. Each data point reported below is an average of 30 simulation runs.

In the first set of simulations, we compare the performance of the full-fledged version of the proposed algorithm, including the distance calibration and tabulation procedure and the two phase CH volunteering procedure, against two partial versions of the proposed algorithm and the static clustering approach. The first partial version includes only one phase in CH volunteering, i.e., each time a CH intends to volunteer itself as the leader, it sends a complete signature packet after its delay timer expires. Also, the first version does not construct the tables in advance to facilitate determination of the back-off timer values. Instead a CH determines its backoff timer value based on Eq. (3), and a sensor sets its backoff timer in proportion to the ratio of the signal strength detected by the soliciting CH to that detected by itself. The second partial version employs the two phase CH volunteering procedure but does not exercise the distance calibration and tabulation procedure. In the simulation, the target generates an acoustic event every 0.5 second. The target moves continuously

⁵If the slot time is 20 μ second, W_{ran} can be divided into 5 slots.

at the speed of maximum 20 m/s under the random waypoint model. As each simulation run lasts for 1000 seconds, at most 2000 events can be detected.

Table 2 gives the comparison results under the first configuration (square deployment). Under the assumption of negligible noises, each sensor can precisely estimate the distance from the target to itself based on the magnitude of its received signal strength. As shown in Table 2, the proposed approach incurs the minimum location error, the smallest latency, and the least amount of message exchange. The static clustering approach incurs the largest location error, as a CH that is not in the vicinity of the target may be responsible for estimating the location and reporting the event. Without the two-phase volunteering procedure, the first partial version of the proposed algorithm incurs significant collision. This is in part because once a solicitation packet is handed over to the MAC layer for transmission, it cannot be cancelled by the high-layer algorithm even if collision occurs. In the case that the solicitation packet contains both the signal strength and the signature, the chance for MAC level collision increases. This problem is notably mitigated in the second partial version. However, without carefully setting the backoff timer values, the second partial version still incurs heavy message overhead. In contrast, the full-fledged version incurs the least collision and the least amount of message exchange, the former due to the two-phase CH volunteering procedure, and the latter due to the use of Voronoi diagram to properly determine the backoff timer values. In the next set of simulations, we investigate the impact of noise on the performance of the proposed algorithm. The noise for each sensor, w_i in Eq. (1), is the product of noise magnitude and additive Gaussian distribution $N(0, 1.0)$. Fig. 5(a) gives the average location error and latency under different magnitudes of noise. The noise with magnitude 40 is approximately equivalent to SNR = 10. The proposed algorithm performs much better than the static clustering algorithm. This is partially because the proposed algorithm has figured in the effect of possible errors on measurements and can tolerate certain levels of noise. Next, we investigate the effect of the moving speed of the target on the performance (with the noise magnitude fixed at 40). As shown in Fig. 5(b), even though the performance of the proposed algorithm degrades as the moving speed of the target increases, it is still the most robust algorithm as compared to other three approaches. Finally we investigate the performance of the proposed algorithm under the assumption that both CHs and sensors are uniformly distributed within the detection range in the area. Fig. 6(a) gives the results of the full-fledged and partial versions of the proposed algorithm under nine deployment configurations with noise = 40 and maximum speed = 20 (m/s). The full-fledged version of the proposed approach outperforms two other partial versions, although its performance also degrades. This is due to the fact that as CHs are

not deployed uniformly, under certain cases none of the CHs detects the event. Another interesting observation is that the gap between the first and second partial versions is significant. This demonstrates the usefulness of the two phase CH volunteering procedure. Fig. 6(b) gives the average performance under the nine deployment configurations as the moving speed of the target varies. The result exhibits similar trends to that under square deployment.

VI. CONCLUSIONS

In the paper, we devise and evaluate a fully decentralized, light-weight, dynamic clustering algorithm for target tracking. We envision a hierarchical sensor network that is composed of (a) a static backbone of sparsely placed CHs that assume the role of leaders upon triggered by certain signal events; and (b) moderately to densely populated low-end sensors whose function is to provide sensor information to CHs upon request. A cluster is formed and a CH becomes active, when the acoustic signal strength detected by the CH exceeds a pre-determined threshold. The active CH then broadcasts an information solicitation packet, asking sensors in its vicinity to join the cluster and provide their sensor information. We address and devise solution approaches to the issues (II)–(I4) as outlined in Section 1. Through both probabilistic analysis and *ns-2* simulation, we show with the use of Voronoi diagram, the CH that is closest to the target is (implicitly) selected as the leader and that the proposed dynamic clustering algorithm effectively eliminates contention among sensors and renders more accurate estimates of target locations.

We have identified several research avenues. First we will engage in devising a more accurate localization method to deal with sensing errors and multiple tracking targets. Second, we will integrate dynamic clustering with *information quality driven routing* protocols. In conjunction with all the above research tasks, we are currently working on a prototype implementation on a lab testbed that consists of Berkeley motes integrated with PC 104 boards.

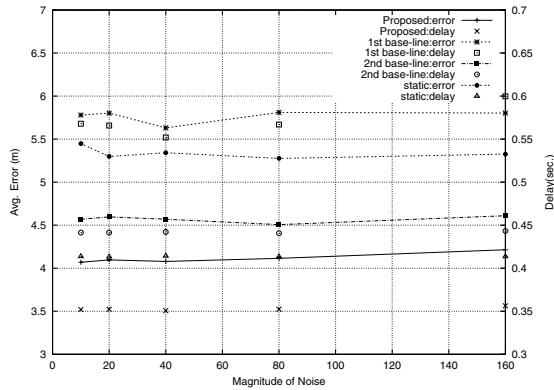
REFERENCES

- [1] F. Aurenhammer, "Voronoi Diagrams - A Survey Of A Fundamental Geometric Data Structure," *ACM Computing Surveys* 23, pp. 345- 405, 1991.
- [2] M. Chu, H. Haussecker and F. Zhao, "Scalable Information-driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," *Int'l J. High Performance Computing Applications*, vol. 16, no. 3, Fall 2002.
- [3] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proc. 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.
- [4] D. Li, K. Wong, Y. Hu and A. Sayeed, "Detection, Classification, Tracking of Targets in Micro-sensor Networks," *IEEE Signal Processing Magazine*, pp. 17-29, March 2002
- [5] X. Sheng, and Y-H Hu, "Energy Based Acoustic Source Localization," *Proc. of 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.

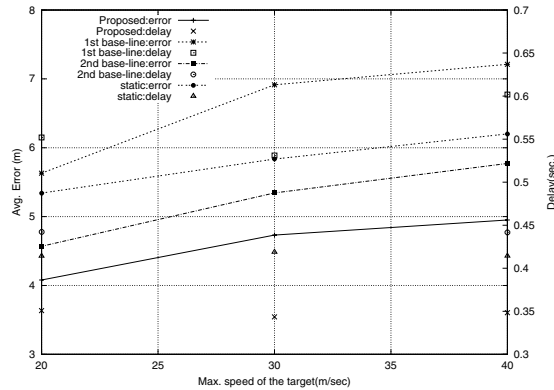
	avg err(m)	latency(s)	detected events	collision times	tot msg sent
static cluster	5.57	0.42	1926	8574	29652
1st base-line	5.89	0.57	1980	30634	57505
2nd base-line	4.87	0.45	1998	10664	40824
proposed algorithm	4.35	0.33	1989	464	17513

TABLE II

COMPARISON BETWEEN THE FULL-FLEDGED VERSION OF THE PROPOSED ALGORITHM, TWO PARTIAL VERSIONS OF THE PROPOSED ALGORITHM, AND THE STATIC CLUSTERING ALGORITHM UNDER SQUARE DEPLOYMENT AND THE ASSUMPTION THAT NOISE CAN BE IGNORED.

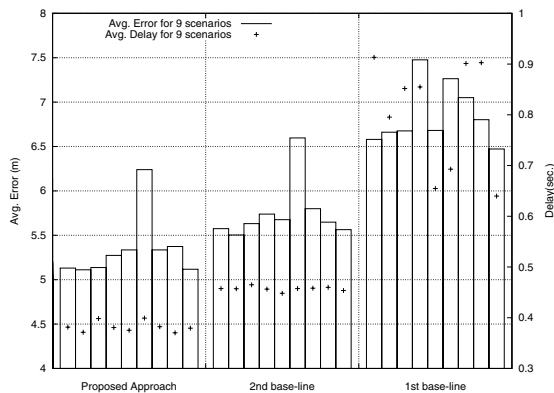


(a) Performance under different magnitudes of noise

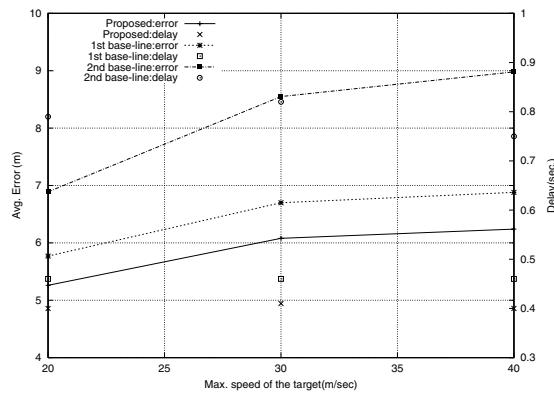


(b) Performance under different moving speeds of the target

Fig. 5. Performance of the proposed algorithm under different magnitudes of noise ((a)) and different moving speeds of the target ((b)).



(a)



(b)

Fig. 6. Performance of the full-fledged and partial versions of the proposed algorithm under nine different deployment configurations with noise = 40 and maximum speed = 20 ((a)) and with noise = 40 and varying moving speeds ((b)).

[6] Q. Wang, W-P Chen, R. Zheng, K. Lee, and L. Sha, "Acoustic Target Tracking Using Wireless Sensor Devices," *Proc. of the 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.

[7] F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang, "A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks," *ACM International Conference on Mobile Computing and Networking (Mobicom 2002)*, Atlanta, Georgia, September 2002.

[8] F. Zhao, J. Shin and J. Reich, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications," *IEEE Signal Processing Magazine*, March 2002.

[9] UCB, LBNL, "VINT network simulator," <http://www-mash.cs.berkeley.edu/ns/>.

[10] UCLA, "SensorSim : A Simulation Framework for Sensor Networks," <http://nesl.ee.ucla.edu/projects/sensorsim/>.

[11] C. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal of Selected Areas in Communications*, Vol. 15, No.7, September 1997

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *Proc. Hawaiian Int'l Conf. on Systems Science*, January 2000

[13] P. Krishna, N. H. Vaidya, M. Chatterjee, D. K. Pradhan, "A Cluster-Based Approach for Routing in Dynamic Networks," *ACM SIGCOMM Computer Communications Review*, April 1997

[14] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, August 1999, pages 1395-1414.