Elastic Localization

Pascal A. Vicaire, John A. Stankovic
Department of Computer Science, University of Virginia
pv9f@cs.virginia.edu, stankovic@cs.virginia.edu
Technical Report CS-2004-35

***Abstract*— Numerous wireless sensor network algorithms assume that individual sensors possess location information. However, state of the art localization algorithms often achieve acceptable performance only under restrictive assumptions. For instance, some algorithms necessitate regular sensor deployment or centralized computations. Other algorithms require a high proportion of position aware nodes or the ability to accurately infer emission distance or emission direction of received radio signals.**

We propose the Elastic Localization Algorithm (ELA), a distributed, scalable, robust and efficient localization algorithm. ELA only presumes that a few percent of the sensors know their location and that an estimation of the maximum communication range is available. We provide extensive simulation data describing the precision, the convergence speed, and the communication load of ELA, using networks composed of thousands of sensors. In addition, we submit ELA to testing considering the influence of maximum range and beacon position misestimation, irregular radio patterns, asynchronous nodes, packet losses, particular topologies, and sensor mobility.

I. Introduction

Wireless sensor networks (WSNs) have attracted much interest as a means to monitor and control the physical world. Many innovative applications of such networks have been proposed including environmental and biological monitoring, target tracking, and smart environments.

Naturally, location information is essential to such applications: tracking algorithms report the position of a target over time, environment monitoring networks inform of the details of relevant events and particularly of where and when they occurred. Consequently, localization algorithms constitute an essential part of WSN research.

The WSN literature contains numerous localization schemes aiming to provide each sensor with accurate location estimation. Many such schemes necessitate expensive and energy consuming hardware. Rather than relying on such additional hardware, we proceed under minimum assumptions: we only assume that a small fraction of the sensors, called beacons, know their location and that sensors possess an estimation of their maximum radio range.

This paper makes the following contributions to the localization problem in WSNs. First, we present a novel equation to more accurately estimate the distance between two sensors, based on hop-count. Second, we exploit, in depth, an analogy with a system of masses linked by springs to efficiently evaluate sensor positions. Third, we demonstrate a new way to use distance estimations not only to beacons, but also to nearby sensors to enhance localization accuracy. Fourth, we show how a multiphase algorithm can further improve performance by first roughly and then precisely locating sensors. Fifth, as an integration of the previous concepts, we propose ELA, a robust, efficient and accurate localization algorithm.

In addition, this paper analyzes the performance of ELA through extensive simulation. We present results according to six different metrics characterizing ELA accuracy and communication overhead. We also provide scalability results for networks of more than 25,000 nodes. Furthermore, we thoroughly evaluate the influence of density and beacon proportion on performance. We then test ELA on four kinds of randomly generated topologies and on regular grids. We then evaluate the robustness of our algorithm with respect to five phenomena occurring in real networks such as asymmetric links and packet loss. We subsequently compare ELA to three state of the art algorithms. Finally, we investigate the performance of ELA in mobile networks. Our results show that ELA offers significant improvement over current solutions.

The remainder of this paper is organized as follows. Section II discusses previous work. Section III describes our algorithm in detail. Section IV thoroughly evaluates the performance of ELA and compares it to state of the art algorithms. Finally, Section V concludes the paper.

II. Related Work

In this section, we discuss previous solutions for providing location information to individual sensors. We successively discuss methods based on time of arrival, time

difference of arrival, received signal strength, angle of arrival, high radio range beacons, and hop-count. We do not discuss solutions using infrared [1] or magnetic fields [2]. They necessitate complex infrastructures surrounding the sensors, are very expensive, and of limited scalability.

Time of arrival techniques rely on the evaluation of signal propagation time to infer location information. Examples of such systems include the global positioning system (GPS). GPS [3] necessitates expensive and energy consuming hardware. Moreover, GPS devices need to be in direct line of sight with at least three satellites and are therefore unusable indoors.

Time difference of arrival techniques infer location in the same way as bats. Some dedicated devices emit radio signals and wait for replies. The time difference between solicitation emission and reply reception permits sensors to evaluate the distance between requesting and answering devices. Such systems either require complex infrastructures [4] [5] or expensive and energy consuming embedded hardware [6]. Techniques using ultrasound signals such as [4] have another disadvantage: they need a high density of sensors to compensate for the poor radio range of ultrasonic signals.

Techniques relying on received radio signal strength assume the existence of a relation between received radio power and relative distance of the emitting sensor ([7], [8], [9]). We first note that such a relation may heavily depend on environmental conditions such as rain, presence of obstacles, irregular signal propagation or interferences. Additionally, such techniques require additional hardware that may not be available on all platforms.

Angle of arrival methods allow each sensor to evaluate the relative angles between received radio signals. The main disadvantage of such system is the additional hardware they employ [8]. Furthermore, current hardware does not provide enough information to infer 3D location, thereby limiting localization to a 2D space.

Solutions using high radio range beacons assume that these beacons possess higher radio range than regular sensors. Thereby, they can reach more sensors, which can consequently compute their location more accurately. This method requires that beacons possess more powerful radio transmitters and higher reserves of energy. Two such algorithms are Bulusu's solution [10] and APIT [11]. We compare ELA to APIT in Section IV-E.

Hop-count algorithms do not need any extra hardware. Beacons disseminate location information through the network and sensors estimate their position from hop-count. ELA pertains to this particular category. In Section IV-E, we compare ELA to two state of the art hop-count algorithms: DV-Hop [12] and Nagpal's solution [9]. MCL [13] also pertains to the class of hop-count algorithms. It is particularly suited to mobile networks and uses Monte Carlo filtering techniques to iteratively evaluate sensor positions. We compare ELA to MCL in Section IV-F.

III. Elastic Localization Algorithm

Let us consider a system of heavy masses, linked by springs of various equilibrium length, confined in a vast room. Let us assume that the equilibrium length of a spring linking two sensors is equal to the physical length separating the two sensors. Then, let us ask some operators to push all the masses towards the center of the room except some particular masses, called anchors, that are fixed to the floor and thus unmovable. Once all the masses reached the center of the room, let us simultaneously release all of them. Assuming that springs and masses do not hamper each other, we can intuitively imagine that the masses regain their initial position because of the spring forces acting upon them. At least, we can believe that the more anchors and springs the system contains, the closer to their initial position the masses end up.

ELA proceeds from an analogy with such a system of masses. Sensors play the role of masses. Sensors that know their locations, which we name beacons, play the role of anchors. If a sensor knows about the existence of a neighbor, it behaves as if it is linked to it by a spring. Sensors possess a physical location and a virtual location. The physical location is unknown from the sensor. The virtual location is initially set to the cartesian coordinates value $(0, 0)$. In the analogy, this is equivalent to gather all the masses in the center of the room. Then, sensors periodically broadcast their virtual position, the virtual position of their neighbors and the physical position of their closest beacons. Receiving this information, sensors can iteratively identify neighboring sensors and beacons and compute how many communication hops separate them from these other devices. Considering their current virtual position, sensors can deduce that they should be located further or closer to a given neighbor or beacon and modify their virtual location accordingly. In the system of masses and springs, the instantaneous force exerted by a given mass on another one plays the role of the sensor broadcast. Indeed, a mass moves to satisfy the constraints imposed by the repulsive and attractive forces acting upon it. The system of masses reaches equilibrium when all the masses regained their initial position. At this point, the length of a given spring equals its equilibrium length and there is no more forces acting on the masses. Similarly, sensors reach equilibrium when their virtual position approximates well their physical position. At

this point, broadcasted messages only indicate to a given sensor that it is at an adequate virtual distance from its neighbors. Note that we need a way to evaluate the distance between two sensors according to the minimum number of communication hops separating them.

In the remainder of this section, we describe in detail ELA. Sensors only know their maximum radio range r_{max}. Additionally, a small fraction of sensors, the beacons, know their exact location. They may obtain this information through a GPS receiver or manual programming. We emphasize that beacons possess the same radio range r_{max} as other sensors and henceforth communicate exclusively with sensors physically located at a distance d such that $d \leq r_{max}$.

We define the hop distance between two sensors A and B, noted $d_{hop}(A, B)$ as the smallest number of communication hops necessary for B to contact A. We note that there may exist several shortest communication paths (in terms of communication hops) between two given sensors. We also emphasize that in the presence of asymmetric links, $d_{hop}(A, B)$ may differ from $d_{hop}(B, A)$. An N-hop neighbor of a sensor A is a sensor B such that $d_{hop}(A, B) = N$. A one-hop neighbor is also called an immediate neighbor. The physical distance between two sensors A and B is noted $d(A, B)$. The average hop length between two sensors A and B is defined as $l_{hop}(A, B) = d(A, B)/d_{hop}(A, B)$. Sensors initially believe their position to be $\mathbf{x}_s(0) = (0, 0)$.

In the following subsections, we describe the four main components of ELA: the ELA datagram, the ELA approximation, the spring equations, and the three-phase protocol. In order to infer their position, sensors need to exchange information with their immediate neighbors. They communicate using a particular message format called the ELA datagram. By periodically communicating, sensors acquire data about one-hop neighbors, two-hop neighbors and nearby beacons. They subsequently use the ELA approximation to estimate their relative distance to these neighbors and beacons. Sensors then individually combine this approximate information to compute a new position estimation. To achieve that goal, each sensor considers itself to be linked to neighbors and beacons with springs and attempts to reach an equilibrium within the so formed system of springs. One problem with systems of masses linked by springs is that they possess numerous equilibrium configurations depending on their initial conditions. Our three-phase protocol aims to make the system of springs converge towards an equilibrium corresponding to accurate estimations of sensor physical positions.

TABLE I

FIELD DESCRIPTION OF THE ENTRIES ENCOUNTERED IN THE THREE TABLES OF A SENSOR S.

Field Name	Field Description
Sensor ID	Unique ID of immediate neighbor (one-hop table entry), two-hop neighbor (two-hop table entry), or beacon (beacon table entry).
Estimated Position	Last advertised position of immediate neighbor (one-hop table entry), two-hop neighbor (two-hop table entry), or physical position of beacon (beacon table entry).
Virtual Speed	Difference between last position estimation and next to last position estimation of immediate neighbor (one-hop table entry) or two-hop neighbor (two-hop table entry). A beacon table entry does not contain any such a field. We arbitrarily consider the time interval between two position estimations to be one second.
Stage	Current stage of immediate neighbor i (see three-phase protocol). Only a one-hop table entry contains such a field.
Cumulative Number of Neighbors	Number of neighbors of datagram sender i plus number of neighbors of datagram receiver s plus number of neighbors of each sensor on a minimal path from i to s.
Hop Distance	Current estimation of hop distance between beacon i and sensor s (beacon table entry only).

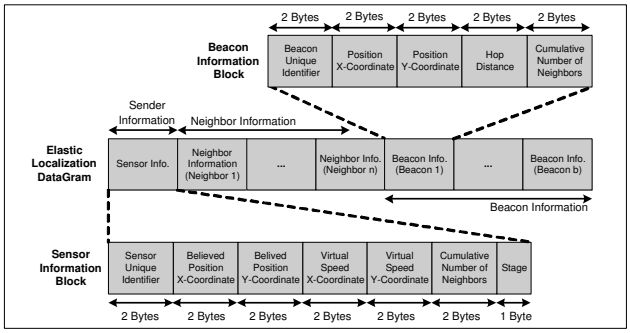

Fig. 1. Elastic localization datagram.

A. Elastic Localization Datagram

Each sensor maintains a one-hop table, a two-hop table and a beacon table, which respectively contain information about one-hop neighbors, two-hop neighbors and beacons. Table I describes the fields encountered in entries of these tables. The three tables are initially empty. Periodically, sensors broadcast information concerning themselves, one-hop neighbors and beacons. They exclusively broadcast information that has changed since the last datagram emission. They compact the data into an ELA datagram, of which Figure 1 reveals the structure. To save space, we do not detail the neighbor information block in the figure. It is identical to the sensor information block except that it does not include a stage field.

Upon receiving a datagram, sensors accordingly complete their three tables. The datagram provides information about the datagram sender, the n one-hop neighbors known of the datagram sender, and the b beacons for

which the datagram sender currently saves information. Sensors only retain information about their b_{max} closest beacons. In the remainder of this paper, we impose $b_{max} = 40$: this experimentally discovered value leads to a good overall performance.

Sensors create or modify one-hop, two-hop and beacon table entries, according to, respectively, the sender, the neighbor, and the beacon information contained in the received datagram. If the neighbor information concerns a sensor already in the one-hop table or the receiver itself, the information is discarded. Completing their three tables, sensors iteratively compute the hop distance d_{hop} to their b_{max} closest beacons, and the cumulative number of neighbors n_c on shortest communications paths to neighbors and beacons. We define n_c as the sum of the number of neighbors of the information sender i, of the information receiver s, and of each sensor on a minimal path from i to s. From n_c and d_{hop}, sensors can deduce n_a, the average number of neighbors on one of the shortest communication paths, using the formula $n_a = n_c/(d_{hop}+1)$. Subsequently, sensors use n_a and d_{hop} to estimate their physical distance to neighbors and beacons. Finally, sensors use this physical distance estimation to adjust their location as described in Sections III-C and III-D.

B. Elastic Localization Approximation

In the previous section, we saw how sensors obtain an estimation of the hop distance d_{hop} to their closest beacons as well as the average number of neighbors n_a on paths to one-hop neighbors, two-hop neighbors and beacons. Sensors also know their maximum radio range r_{max}. The goal of this section is to empirically evaluate the average physical distance, here noted d, from a given sensor to each of its neighbors or beacons with respect to d_{hop}, n_a and r_{max}. Previous research [14] demonstrates that n_a and r_{max} play an important role in the determination of d. Our introduction of d_{hop} derives from intuition. To empirically evaluate d according to the specified parameters, we simulate huge networks of randomly distributed sensors. For each value of n_a varying from one to thirty, we generate twenty networks of 5000 sensors. In each network, we compute d_{hop} from each sensor to any other one. For given values of d_{hop} and n_a, we finally evaluate the average d. The data resulting from these simulations allow us to derive the ELA approximation, an approximation of the average

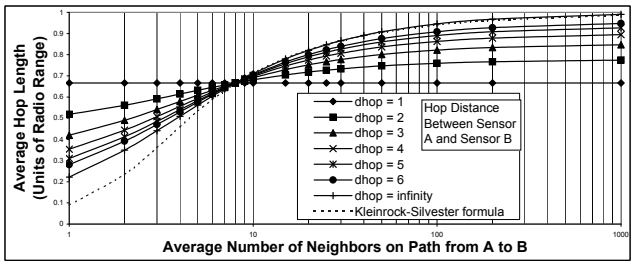

Fig. 2. ELA approximation and Kleinrock-Silvester formula.

value of $l_{hop} = \frac{d}{d_{hop}}$ as a function of d_{hop}, n_a, and r_{max}:

$$\frac{l_{hop}}{r_{max}} = \left(1 - \left(\frac{2}{3}\right)^{d_{hop}-1}\right) \frac{2\arctan\left(\left(\frac{n_a}{8}\right)^{\frac{3}{4}}\sqrt{3}\right)}{\pi} + \left(\frac{2}{3}\right)^{d_{hop}} \quad (1)$$

We obtained this formula by first observing that our experimental curves had a shape similar to that of the $arctan$ function and by subsequently using curve fitting techniques. In comparison, to approximate l_{hop}, Nagpal et al. [9] use the Kleinrock-Silvester formula [14] given as:

$$\frac{l_{hop}}{r_{max}} = 1 + e^{-n_a} - \int_{-1}^{1} e^{-\frac{n_a}{\pi}\left(\arccos t - t\sqrt{1-t^2}\right)} dt \quad (2)$$

The Kleinrock-Silvester formula is an approximation of the average value of l_{hop} as a function of r_{max} and n_a. Figure 2 graphs the ELA approximation and the Kleinrock-Silvester formula. The empirical data from which we derived Equation 1 indicates that l_{hop} varies significantly with respect to d_{hop}. We therefore concluded that using the ELA approximation (a function of d_{hop}, n_a, and r_{max}) should lead to better performance than using the Kleinrock-Silvester formula (a function of n_a and r_{max}).

In addition, we observe that when d_{hop} approaches infinity, the ELA approximation well approximates the Kleinrock-Silvester formula for $n_a \geq 10$. This could be expected as the Kleinrock-Silvester formula is an approximation of l_{hop} when d_{hop} tends towards infinity. For $n_a < 10$, the ELA approximation when d_{hop} approaches infinity differs significantly from the Kleinrock-Silvester formula. This is due either to the empirical nature of our approximation or to the simplifying assumptions used in [14] to establish Equation 2. Additionally, we note that when $d_{hop} = 1$, $\frac{l_{hop}}{r_{max}} = \frac{2}{3}$. The value $\frac{2}{3}$ adequately corresponds to the theoretical average distance between the center of a circle C of radius 1 and iid randomly chosen points within C. The behavior of the ELA approximation when $d_{hop} = 1$ and when d_{hop} approaches infinity thus reinforces our confidence into the validity of our approximation.

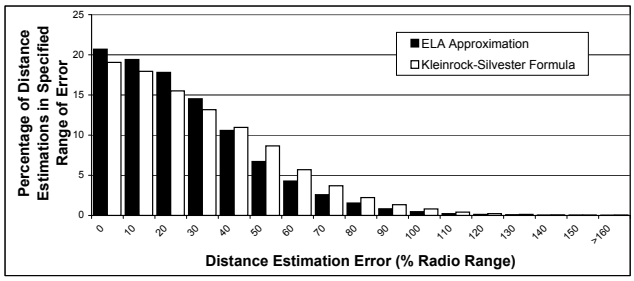

Fig. 3. Comparison of the distribution of distance estimation errors generated by the ELA approximation and the Kleinrock-Silvester formula.

In Figure 3, we compare the performance of the ELA approximation with the performance of the Kleinrock-Silvester formula. We simulate twenty networks of size 5000 with sensors iid randomly distributed within a square area. The average number of neighbors is fifteen and the number of beacons is 500. Each sensor evaluates its distance to its forty closest beacons using successively the ELA approximation and then the Kleinrock-Silvester formula. It subsequently compares distance estimates to real distances. Figure 3 shows that 21% of the distance estimates that the ELA approximation generated had an error within the interval $[0, 10\%r_{max}]$. By comparison, only 19% of the distance estimates that the Kleinrock-Silvester formula provided had an error within the same interval. By observing the error distribution in its entirety, we can clearly state that the ELA approximation performs better than the Kleinrock-Silvester formula.

Sensors can now evaluate the physical distance to their neighbors and beacons. They already know the current position estimations of their neighbors and the physical positions of their b_{max} closest beacons. Each couple $(distance, position)$ imposes a constraint on the sensor, limiting its choices for a new position estimation. In fact, it is likely that no choice of position simultaneously satisfies all the constraints imposed on the sensor. Indeed, sensors initially estimate their position to be $(0, 0)$ and the ELA approximation only provides a gross approximation of the distance between two sensors.

To re-estimate their position, sensors imagine that they are linked to their neighbors and beacons with springs. We determine the equilibrium length of each spring using the ELA approximation. Each sensor iteratively moves towards one of the equilibrium positions defined by its system of springs. Between iterations, it receives new estimations of neighbor's positions. In this way, we expect sensors to iteratively discover an accurate estimation of their real position. The next two sections describe the computation of new position estimations using spring equations. We first discretize spring equations inspired from physics before modifying them to ensure quick convergence.

C. Discretized Spring Equation

We now consider that sensors possess an estimation of their physical distance to neighbors and beacons. These estimations are obtained through the ELA approximation as explained in the previous section. In this section, sensors re-evaluate their position according to this information. They do so each time they receive a datagram. We proceed with an analogy with the physics of springs. A given sensor s behaves as if it were a mass linked with springs to known neighbors and beacons. According to Newton's Second Law, s is subject to an acceleration $\ddot{\mathbf{x}}_\mathbf{s}$ such that:

$$m\ddot{\mathbf{x}}_s(t) = \sum \mathbf{F}(t) \tag{3}$$

Where m is the mass of sensor s, $\sum \mathbf{F}$ is the sum of external forces acting on s, and t is the current time. We can arbitrarily impose $m = 1kg$ to simplify the equations. According to Hooke's law, a spring linking sensor s to another sensor i exerts on s a force $\mathbf{F}_{\mathbf{i}\rightarrow\mathbf{s}}$ such that:

$$\mathbf{F}_{i\rightarrow s}(t) = -k\left(\|\mathbf{x}_s(t) - \mathbf{x}_i(t)\| - d_i\right)\frac{\mathbf{x}_s(t)-\mathbf{x}_i(t)}{\|\mathbf{x}_s(t)-\mathbf{x}_i(t)\|} \tag{4}$$

Where k is the spring constant, $\mathbf{x_i}(t)$ is the physical position of sensor i at time t, $\mathbf{x_s}(t)$ is the physical position of sensor s at time t, d_i is the equilibrium length of the spring that links sensor s to sensor i, and $\|\mathbf{a}\|$ denotes the norm of vector $\mathbf{a}$. We can arbitrarily impose $k = 1Nm^{-1}$. If $\mathbf{x}_s(t) = \mathbf{x}_i(t)$, we apply a random displacement of unitary norm to sensor i. Figure 4 illustrates Equation 4. We modify Equation 4 such that $\mathbf{x_i}(t)$ is the estimation by sensor i of its position at time t, $\mathbf{x_s}(t)$ is the estimation by sensor s of its position at time t, and $d_i = d_i(t)$ is an estimation of the physical distance between sensor i and sensor s as computed in Section III-B. Additionally, we need to discretize Equation 4, estimating the values of physical variables at time t as a function of the values of physical variables at time $(t-1)$. We assume that sensors broadcast their information simultaneously and instantly with an identical period T. We relax this assumption in Section IV-D. We can arbitrarily impose $T = 1s$. If i is a one-hop neighbor of s, we evaluate $\mathbf{F}_{i\rightarrow s}^{1hop}$ in the following way:

$$\begin{aligned}\mathbf{F}_{i\rightarrow s}^{1hop}(t) = \; & -\left(\|\mathbf{x}_s(t-1) - \mathbf{x}_i(t-1)\| - d_i(t-1)\right) \\ & \times \frac{(\mathbf{x}_s(t-1)-\mathbf{x}_i(t-1))}{\|\mathbf{x}_s(t-1)-\mathbf{x}_i(t-1)\|}\end{aligned} \tag{5}$$

If sensor i is a two-hop neighbor of sensor s, we cannot directly use Equation 5 because $\mathbf{x}_i(t-1)$ is unavailable.

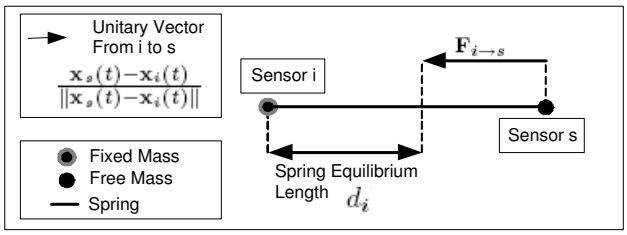

Fig. 4. Illustration of Hook's law.

Nevertheless, we can evaluate $\mathbf{x}_i(t-1)$:

$$\mathbf{x}_i(t-1) = \mathbf{x}_i(t-2) + T\dot{\mathbf{x}}_i(t-2) \tag{6}$$

Where $T = 1s$, $\mathbf{x}_i(t-2)$ is the last received position estimation of sensor i, and $\dot{\mathbf{x}}_i(t-2)$ is the virtual speed of sensor i at time $(t-2)$ defined as in Table I. We can now evaluate $\mathbf{F}_{i\to s}^{2hop}$:

$$\begin{aligned}\mathbf{F}_{i\to s}^{2hop}(t) = & -(\|\mathbf{x}_s(t-1) - \mathbf{x}_i(t-2) - \dot{\mathbf{x}}_i(t-2)\| - d_i(t-1)) \\ & \times \frac{\mathbf{x}_s(t-1)-\mathbf{x}_i(t-2)-\dot{\mathbf{x}}_i(t-2)}{\|\mathbf{x}_s(t-1)-\mathbf{x}_i(t-1)-\dot{\mathbf{x}}_i(t-2)\|}\end{aligned} \tag{7}$$

If i is a beacon, $\mathbf{F}_{i\to s}^{beacon}$ is easy to evaluate as beacons know their position and are immobile:

$$\begin{aligned}\mathbf{F}_{i\to s}^{beacon}(t) = & -(\|\mathbf{x}_s(t-1) - \mathbf{x}_i\| - d_i(t-1)) \\ & \times \frac{\mathbf{x}_s(t-1)-\mathbf{x}_i}{\|\mathbf{x}_s(t-1)-\mathbf{x}_i\|}\end{aligned} \tag{8}$$

We can now evaluate all the spring forces exerted on a given sensor s. To eliminate oscillations in our system composed of springs and masses, we consider that each sensor s is additionally subject to a friction force $\mathbf{F}_{friction}$ such that:

$$\mathbf{F}_{friction}(t) = -f\dot{\mathbf{x}}_s(t-1) \tag{9}$$

Where we impose $f = 1kgs^{-1}$ to obtain maximum friction: with such a value, the sensor remains immobile when no forces are applied to it. Finally, we obtain $\ddot{\mathbf{x}}_s(t)$, $\dot{\mathbf{x}}_s(t)$, and $\mathbf{x}_s(t)$:

$$\begin{aligned}\ddot{\mathbf{x}}_s(t) &= \textstyle\sum \mathbf{F}_{i\to s}^{1hop}(t) + \sum \mathbf{F}_{i\to s}^{2hop}(t) + \sum \mathbf{F}_{i\to s}^{beacon}(t) \\ &\quad + \mathbf{F}_{friction}(t) \\ &= \textstyle\sum \mathbf{F}_{i\to s}^{springs}(t) - \dot{\mathbf{x}}_s(t-1) \\ \dot{\mathbf{x}}_s(t) &= \dot{\mathbf{x}}_s(t-1) + T\ddot{\mathbf{x}}_s(t) = \textstyle\sum \mathbf{F}_{i\to s}^{springs}(t) \\ \mathbf{x}_s(t) &= \mathbf{x}_s(t-1) + T\dot{\mathbf{x}}_s(t) \\ &= \mathbf{x}_s(t-1) + \textstyle\sum \mathbf{F}_{i\to s}^{springs}(t)\end{aligned} \tag{10}$$

D. Spring Equation Convergence

One problem with Equation 10 is that it does not ensure the convergence of the algorithm. In fact, if ELA only uses Equation 10, the sensor position estimations do not converge because of the discretization of the spring equation. For ELA to be a viable localization algorithm, it must converge and converge quickly. This section describes the computation of a value of $\mathbf{x}_s(t)$ different from the one proposed in Equation 10, even though inspired by it, to provide quick convergence of ELA. We denote this distinct value of $\mathbf{x}_s(t)$ by $\mathbf{x}_s^c(t)$. We do not provide formal proof of the quick convergence of ELA using the convergent spring equation. However, our simulation data provides strong empirical support in favor of our convergence assumption. We pose:

$$\mathbf{x}_s^c(t) = \mathbf{x}_s^c(t-1) + \tfrac{2}{5}(\mathbf{x}_s^{eq}(t) - \mathbf{x}_s^c(t-1)) \tag{11}$$

Where $\mathbf{x}_s^{eq}(t)$ is the equilibrium position at time t of sensor s in the system composed of springs and masses defined in sensor s at time t. Equation 11, illustrated in Figure 5, makes sensor s move towards $\mathbf{x}_s^{eq}(t)$, which we estimate using the following algorithm:

$$\begin{aligned}&\mathbf{x}_s^{eq}(t) = \mathbf{x}_s^c(t-1); \\ &\texttt{while}\left(\tfrac{1}{N}\|\textstyle\sum \mathbf{F}_{i\to s}^{springs}(t, \mathbf{x}_s^{eq}(t))\| > \tfrac{r_{max}}{\beta}\right)\{ \\ &\quad \mathbf{x}_s^{eq}(t) = \mathbf{x}_s^{eq}(t) + \tfrac{1}{N}\textstyle\sum \mathbf{F}_{i\to s}^{springs}(t, \mathbf{x}_s^{eq}(t));\}\end{aligned} \tag{12}$$

Where r_{max} is the maximum radio range of sensor s, $\sum \mathbf{F}_{i\to s}^{springs}(t, \mathbf{x}_s^{eq}(t))$ is the sum of forces exerted by springs on s computed using $\mathbf{x}_s^{eq}(t)$ instead of $\mathbf{x}_s^c(t-1)$, and N is the total number of spring forces. β is a constant influencing the precision of the equilibrium computation and the convergence of ELA. In the remainder of this paper, we impose $\beta = 1000$: this experimentally discovered value leads to a good overall performance of ELA. We note that during all the experiments realized in Section IV-A, the maximum number of iterations that the loop of Algorithm 12 executed before terminating was 364 and the average number of iterations was equal to thirteen. At each iteration, ELA must evaluate the sum of forces, which is $\theta(n^{1hop}+n^{2hop}+n^{beacons})$ where n^{1hop}, n^{2hop} and $n^{beacons}$ are the number of, respectively, one-hop neighbors, two-hop neighbors, and beacons known of the sensor. The MICA2, a popular WSN platform, can execute several millions of instructions per second and could compute new values of $\mathbf{x}_s^{eq}$ in a timely manner.

E. Three-Phase Protocol

A system of springs and masses can converge towards different final states according to its initial conditions. For instance, Figure 6 represents a system of three masses, $M1$, $M2$ and $M3$, linked by two springs. The equilibrium length of the two springs is identical and equal to the radius of the dashed circles. If we let $M3$ move freely while keeping $M1$ and $M2$ fixed, $M3$ may converge to either

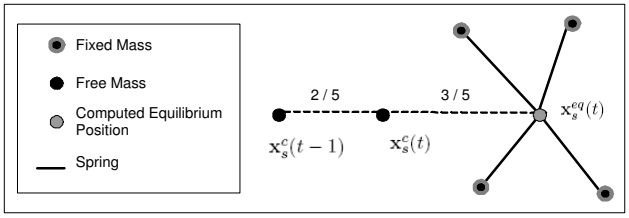

Fig. 5. Movement of sensor s towards equilibrium position.

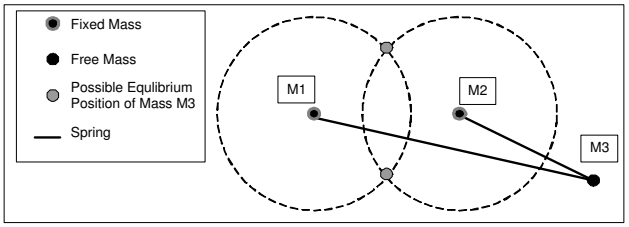

Fig. 6. System of masses linked by springs.

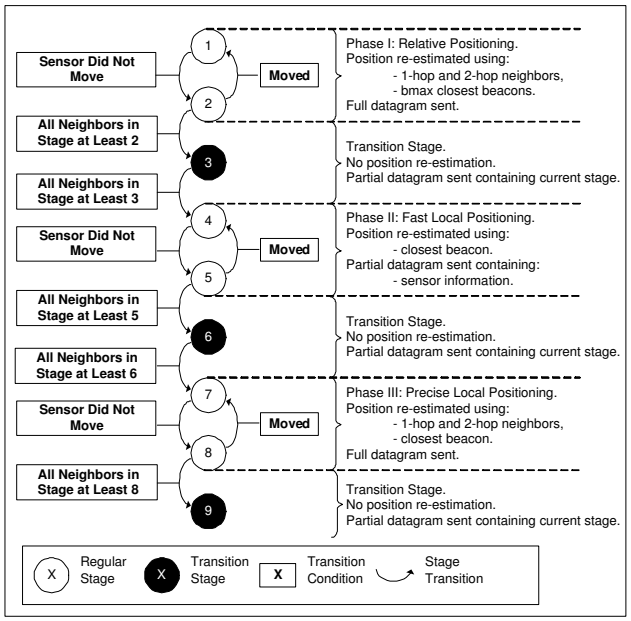

Fig. 7. The nine stages of the three-phase protocol.

one of the gray equilibrium points, depending on its initial position. Similarly, the system of springs and free masses defined in the previous section may converge towards various final equilibrium states. This section presents a three-phase protocol aiming to force sensor position estimations to converge towards a adequate equilibrium, at which position estimations satisfactorily approximate physical positions.

This protocol constitutes the main part of ELA. It contains a subroutine to compute $\mathbf{x}_s^c(t)$ as explained in Section III-D. This subroutine in turn calls a routine to compute d_i as described in Section III-B. The initial stage of a sensor is Stage 1. Sensors periodically broadcast either a full or partial datagram, according to their stage, as described in Figure 7. Each time it receives a datagram from one of its immediate neighbors, a sensor recomputes its position using the spring equation as well as the information contained in this datagram. Comparing its new position estimation $\mathbf{x}_s^c(t)$ to the previous one $\mathbf{x}_s^c(t-1)$, it consequently determines if it has moved. A sensor formally considers itself in motion if and only if:

$$\|\mathbf{x}_s^c(t) - \mathbf{x}_s^c(t-1)\| \geq \frac{r_{max}}{\alpha} \tag{13}$$

Where r_{max} is the maximum radio range of the sensor s, and α is a constant influencing the final accuracy of the position estimation and the convergence speed of the algorithm. For the remainder of this paper, we impose $\alpha = 100$, which leads to a good overall performance of ELA. Once a sensor determines whether it is in motion, it effects a stage transition, if appropriate, as described in Figure 7. It then waits for the next reception of a datagram before re-evaluating its position. Once a sensor reaches Stage 9, it terminates its computations.

The nine stages define three phases called the relative positioning phase (Phase I), the fast local positioning phase (Phase II) and the precise local positioning phase (Phase III). Sensors in Phase I consider to be linked with springs to their one-hop neighbors, two-hop neighbors and b_{max} closest beacons. Thus using all available information, we expect sensors to organize into a global coordinate system where the relativity of estimated positions reflects the relativity of physical positions. One problem with Phase I is that sensors use information from the b_{max} closest beacons. This information is essential to infer sensor relative positions, but may be too imprecise. Indeed, the more hops away a beacon is, the more approximate the estimation of its relative distance is. This is why sensors in Phase II consider themselves to be linked only to their closest beacon: only one spring force contributes to their position re-estimation. Thereby, Phase II aims to adjust sensor position estimation assuming that the estimated distance to the closest beacon is correct. During Phase II, we expect the relativity of sensor positions to be conserved. Finally, sensors in Phase III consider themselves linked not only with their closest beacon, but also with their one-hop and two-hop neighbors. In this way, Phase III attempts to enforce the respect of distance estimations between one-hop and two-hop neighbors.

We have not explained yet why using nine stages while three would seem sufficient: Stage 1, 4 and 7. The reason is that we would like the whole network to compute in a

given phase until convergence before any sensor switches to the next phase. Checking that the whole network has converged is a costly operation. We thus compromise by making each sensor check for the convergence of its neighbors before switching phases. This is the purpose of Stage 2, 5 and 8. Stage 3, 6 and 9 are transition stages. Sensors in a transition stage periodically broadcast information about their current stage and do not re-evaluate their position. They passively wait for neighbors to reach the same stage as theirs. They ensure that a given sensor communicates only with a sensor in the same phase as itself or with a sensor in a transition stage, which is seen as an immobile sensor because it does not re-estimate its position. Thereby, the executions of Phase I, II and III are isolated from one another.

To conclude this section we present accuracy results when using various combinations of phases and forces (see Figure 8). Each data point represents the average value of ten trials using randomly generated topologies of 5000 sensors, including ten beacons. The average number of neighbors is ten. Among the shown variants, our three-phase protocol employing the three forces described in Section III-C generates the least mean absolute error on position estimation. Our experiments reveal that we can generalize this result to many WSN configurations. We therefore focus on this version of ELA in the evaluation section. Applications requiring very low communication overhead may prefer other versions of the algorithm (see Figure 9). We observe that three phases may induce less communication overhead than two phases: one phase can dramatically diminish the execution time of the subsequent ones. We also predict that using three-hop forces would generate prohibitive communication costs. We note that the time complexity of the three-phase protocol is evaluated via simulation in Section IV-A (see Figure 12).

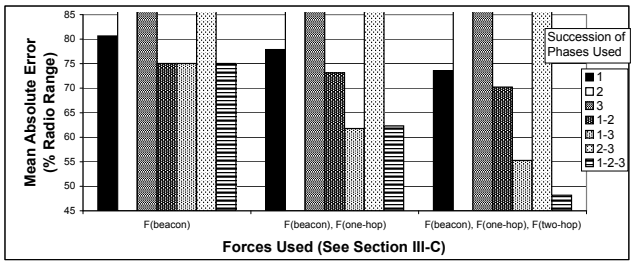

Fig. 8. Influence of phases and forces used on ELA accuracy.

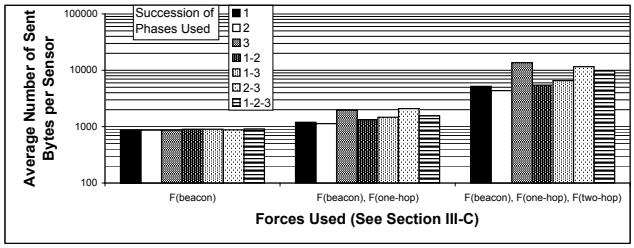

Fig. 9. Influence of phases and forces used on ELA overhead.

IV. Evaluation

This section provides a detailed quantitative analysis of ELA. In all the graphs of this section, each data point represents the average value of ten trials using different randomly generated network topologies. Except if specified otherwise, all sensors (including beacons) are iid randomly distributed within a square area of size $500m \times 500m$. To obtain various average number of neighbors in a given network, we vary the radio range of sensors. Except in Section IV-D, all sensors have identical radio ranges. On the figures, we note NT the number of trials used to obtain one data point, NN the average number of neighbors, NS the number of sensors, and NB the number of beacons. We first investigate the scalability of ELA in Section IV-A. We then study the variation of performance according to the average number of neighbors and the number of beacons in Section IV-B. In Section IV-C, we evaluate the influence of particular topologies on ELA position estimation accuracy. We then furnish evidence of ELA robustness to phenomena occurring in real WSNs in Section IV-D. Finally, we compare ELA to three state of the art algorithms in Section IV-E and consider mobile networks in Section IV-F.

A. Scalability

In this experiment, we observe the influence of the number of sensors and the proportion of beacons on the performance of our algorithm. Sensors possess eleven neighbors on average. Figure 10 shows the Mean Absolute Error (MAE) on position estimation with respect to network size and number of beacons. With ten randomly located beacons, we obtain a MAE of less than 50% of the radio range for networks composed of up to 6000 sensors. At this particular limit, the proportion of sensors is less than 0.17%. With 160 randomly located beacons, we obtain a MAE of less than 25% of the radio range even for networks composed of 25000 sensors (we did not simulate larger networks). At this particular point, the proportion of beacons is 0.64%. For small networks of 200 sensors, ELA performs equally well, obtaining a MAE of about 25% of the radio range when using only five randomly located beacons. We also notice that, in general, bigger networks require a lower proportion of beacons to achieve similar accuracy.

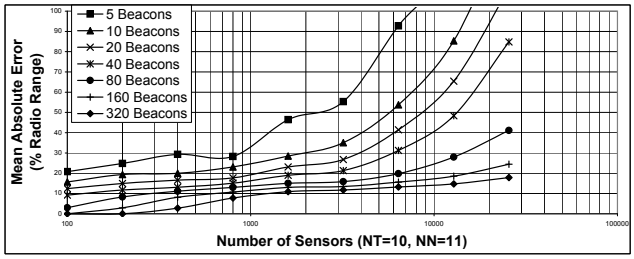

Fig. 10. Scalability results, mean absolute error.

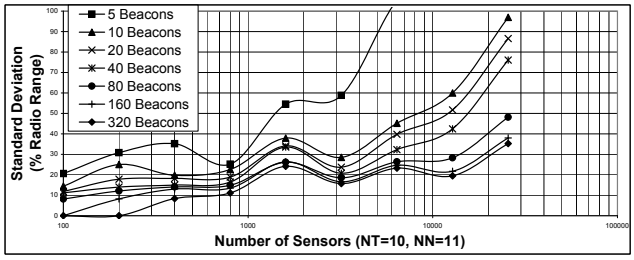

Fig. 11. Scalability results, standard deviation.

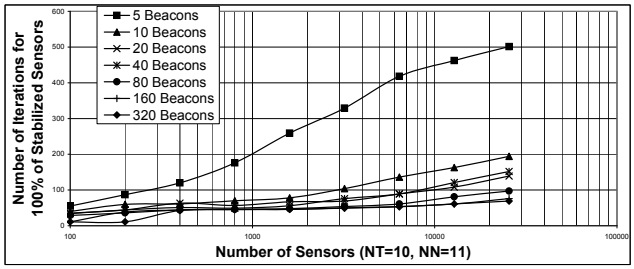

Fig. 12. Scalability results, number of iterations.

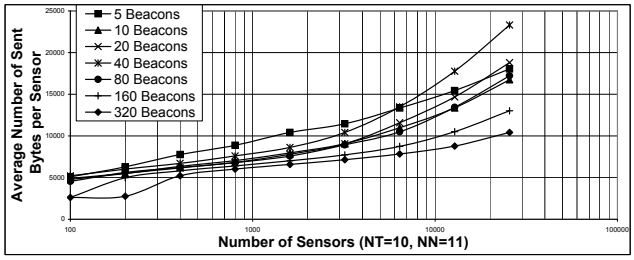

Fig. 13. Scalability results, number of transmitted bytes.

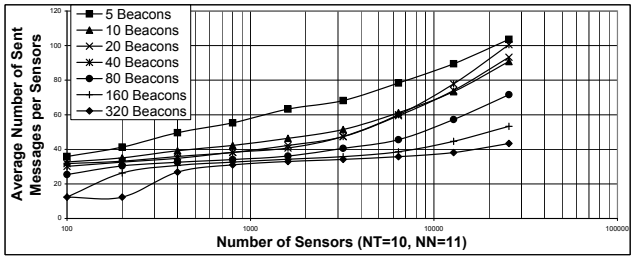

Fig. 14. Scalability results, number of transmitted messages.

Figure 11 exhibits the standard deviation on position estimation error. The results indicate that ELA performs better in some parts of the network than in others. Additional testing predictably revealed that ELA does not perform as well in large parts of the network that are free of beacons.

Figure 12 graphs the number of iterations of ELA necessary for 100% of the sensors to terminate ELA. During an ELA iteration, all sensors that have not completed the algorithm broadcast a datagram, which is eventually processed on neighboring nodes. For small networks of 100 to 200 sensors, only ten to ninety iterations are necessary for complete convergence of ELA. For bigger networks of up to 25000 sensors, more iterations are necessary, but always less than 200 if there are at least ten beacons in the network.

Figure 13 shows the average number of transmitted bytes per sensor when all sensors terminated their computations. We observe that the average number of transmitted bytes varies from about 5000 to 25000 bytes. As an example, the MICA2 mote, a popular WSN platform, achieves a data transmission rate of 38.4 kilobits per second [15]. Assuming such bandwidth, sensors necessitate from about one to five seconds of radio channel availability to complete ELA radio transmissions. We emphasize that in a real WSN, two neighboring sensors cannot hold the communication channel simultaneously. The total time to run ELA is therefore accordingly greater depending on the used media access layer protocol. Additionally, we note that our average number of transmitted bytes considers neither necessary packet headers nor media access layer control messages.

Finally, Figure 13 exhibits the average number of transmitted messages per sensor when all sensors terminated their computations. We observe that in small networks of 200 sensors, less than about forty messages per node are necessary on average. In large networks of 12800 sensors, less than ninety messages per node are on average necessary, even with only five beacons. As the proportion of beacons increases in the network, less and less messages are necessary.

B. Density

In this section, we investigate the impact of the average number of neighbors and the number of beacons on ELA performance. We simulate iid randomly generated networks containing 5000 sensors contained in a square area.

Figure 15 depicts our results. We observe that the accuracy of ELA increases with the average number of

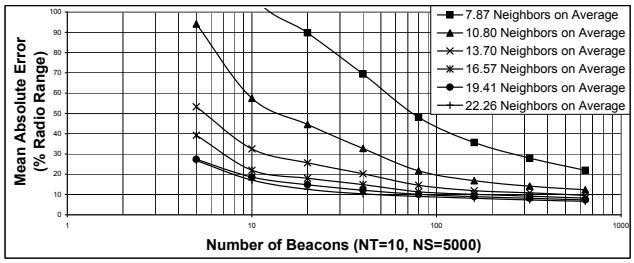

Fig. 15. Influence of density on ELA performance.

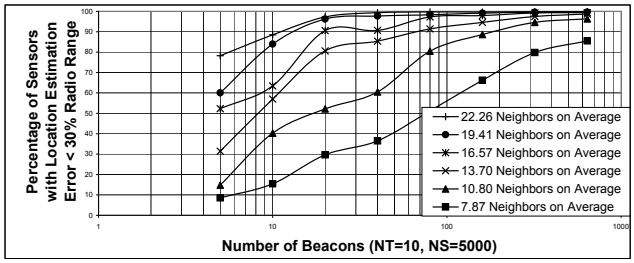

Fig. 16. Influence of density on ELA success.

neighbors and the number of beacons. When using 0.1% of beacons (5 beacons for 5000 sensors), we need an average number of neighbors of about 14 to obtain a MAE of less than 55% of the radio range. When 1% of the sensors are beacons (50 beacons for 5000 sensors), an average number of neighbors of 11 is sufficient to reduce the MAE below 30% of the radio range. When 10% of the sensors are beacons, the MAE is below 25% of radio range even with an average number of neighbors of eight. According to experimental results presented in [11], a localization MAE of 40% of the radio range and an average number of neighbors of eight are sufficient to achieve a delivery ratio of almost 100% when using a routing algorithm such as Geographic Forwarding [16]. This suggests that ELA performance may satisfy the needs of many algorithms that assume the availability of location information.

Figure 16 shows, for the same set of experiments, the percentage of sensors that ELA successfully localized, defining a successfully localized sensor as as sensor localized in a circular area centered at the real location of the sensor and having a radius of 30% of the radio range. We observe that with a proportion of beacons of 0.4% and an average number of neighbors of fourteen, at least 80% of the sensors localize successfully. With the same average number of neighbors and a proportion of beacons of 1.4%, at least 90% of the sensors localize successfully.

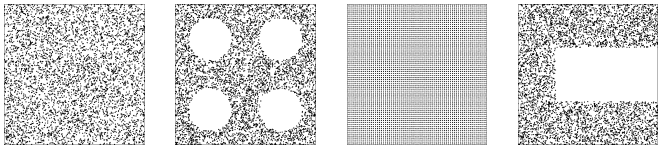

Fig. 17. Example of square topology(a), obstacle topology(b), grid topology(c) and C Shape topology(d).

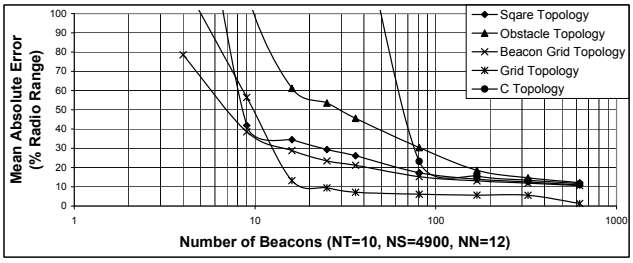

Fig. 18. Influence of topology on ELA performance.

C. Topology

In previous experiments, sensors are randomly distributed within a square. In this section, we study the effect of various topologies on ELA accuracy. We consider five topologies. Figure 17(a) shows a square topology, where sensors are iid randomly distributed within a square. Figure 17(b) shows an example of what we name an ”obstacle topology”, where sensors are iid randomly distributed within a square containing four circular sensor-free zones. Figure 17(c) displays an example of grid topology. Sensor are this time uniformly distributed in a regular grid, with beacons forming a sub-grid. We call ”beacon grid topology” a topology where only the beacons are organized in a grid and where the remaining of the sensors are iid randomly distributed in a square topology. Figure 17(d) presents the C topology, where sensors are iid randomly distributed within an area in the shape of the letter C. Figure 18 graphs the MAE on position estimation according to the number of beacons in the network and the employed topology. All our experiments use an average number of neighbors of twelve and a network of size 4900 sensors.

We first observe the performance of ELA using the square topology. The MAE on position estimation can be as low as 12% of the radio range when 10% of sensors are beacons. Organizing the beacons in a grid significantly increases performance when the total number of beacons is smaller than nine. For a greater number of beacons, the improvement is not as marked. Organizing all the sensors in a grid dramatically ameliorate ELA performance. With only 0.3% of beacons, the MAE is as

low as 10%. And it drops below 3% when the proportion of beacons approaches 10%. By contrast, the existence of obstacles impairs performance as we observe when using the topologies illustrated in Figure 17(b) and 17(d). However, increasing the proportion of beacons to 2% reinstates results similar to the ones obtained with the square topology.

D. Realism

In previous sections, we evaluate the performance of our algorithm assuming perfect conditions: sensors act synchronously, they know their exact maximum radio range, this radio range is identical for all sensors, all communication links are symmetric, beacons perfectly evaluate their location and there is no packet loss. These assumptions are unrealistic in a real sensor network. In this section, we evaluate the performance of ELA in realistic conditions. All the experiments of this section use iid randomly generated square topologies of 5000 sensors with an average number of neighbors of either eleven (Figure 19, 20, 22 and 23) or twelve (Figure 21).

Figure 19 shows the results of an experiment that simulate networks with asynchronous sensors. In the figure, the legend annotation "Period = 1" means that all sensors periodically send datagrams with the same period T. "Period = 1 or 2 or 4" means that one third of the sensors send datagrams with period T, one third of the sensors with period $2T$ and one third of the sensors with period $4T$. Figure 19 shows that sensor asynchrony may significantly degrade performance only when the number of beacons is below 0.25%.

Figure 20 graphs the MAE on position estimation with respect to the number of beacons and the maximum error on radio range estimation. Basically, a maximum radio range error of 25% means that each sensor iid randomly selects a different radio range within the interval $[0.75r_{max}, 1.25r_{max}]$. The graph shows that maximum radio range errors of up to 12% only slightly decrease the accuracy of ELA position estimations. Greater errors more significantly impair performance. However, with 10% of the sensors being beacons, the MAE on position estimation is still below 50% of the radio range even when the maximum radio range error is as high as 50%. Note that the MAE is expressed in percentage of r_{max} and not in percentage of the radio range of the sensors, which is different for each device in this particular experiment.

Figure 21 depicts the results of an experiment testing the influence of asymmetric links on ELA accuracy. When sensor A can send data packets to sensor B and vice-versa, the communication link between A and B is a symmetric

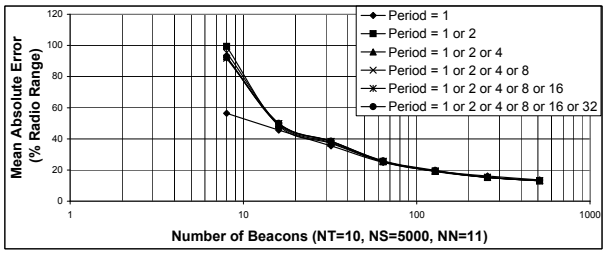

Fig. 19. Influence of sensor asynchrony on ELA performance.

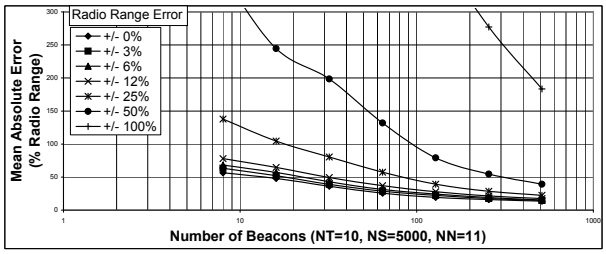

Fig. 20. Influence of radio range error on ELA performance.

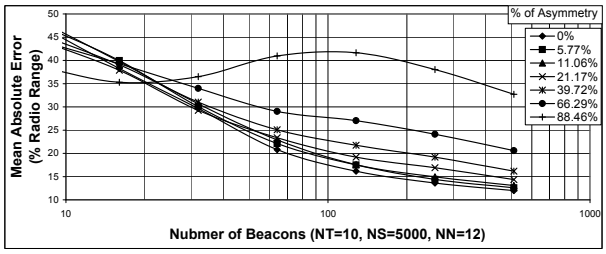

Fig. 21. Influence of asymmetry on ELA performance.

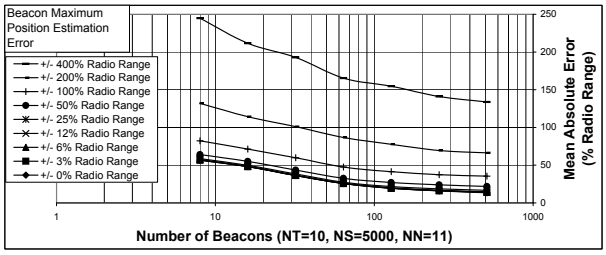

Fig. 22. Influence of beacon position error on ELA performance.

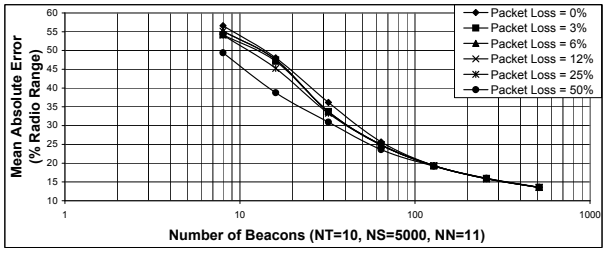

Fig. 23. Influence of packet loss on ELA performance.

link. When only A can reach B, or only B can reach A, the communication link is an asymmetric link. We define the percentage of asymmetry of a wireless sensor network as the percentage of asymmetric links in that network. We emphasize that we keep the average number of neighbors constant during our experiment: when we increase the percentage of asymmetry, we accordingly increase the radio range of sensors. In this experiment, we generated asymmetry by iid randomly removing unidirectional links. We note that this generates not only asymmetry but also radio range irregularity. Figure 21 shows that a percentage of asymmetry below 66% can diminish the MAE on position estimation of at most 15 points. We believe that the percentage of asymmetry in a real wireless sensor network is much lower than 66%, which makes ELA robust to the effect of asymmetry in realistic networks. For reference, all experiments realized in [17] with more than 150 Rene motes indicate a percentage of asymmetric links below 15%.

Figure 22 represents the performance of ELA assuming that beacons do not perfectly know their position. A beacon maximum position estimation error of 25% means that all beacons iid randomly select a different error within the interval $[-25\%r_{max}, +25\%r_{max}]$. The figure shows that the performance of our algorithm is very robust when the beacon maximum estimation error remains below 50% of the radio range. The MICA2 mote achieves an outdoor radio range of 100 ft [15]. Assuming a GPS receiver with an accuracy of 5 meters, we obtain a typical beacon position error of about 16.4% of radio range. ELA thus does not seem to be affected by beacon position errors typical of real networks.

Figure 23 shows the results of experiments evaluating the influence of packet loss on ELA performance. As an example, a packet loss of 50% means that 50% of emitted datagrams are iid randomly dropped by receivers. In this experiment, beacon information is broadcasted reliably while one-hop and two-hop information is subject to packet loss. We expect such packet loss to have little influence on ELA accuracy. Indeed, sensors can work with outdated information until they receive fresh data. Surprisingly, randomly distributed packet loss improves the performance of our algorithm by up to 10 points! We believe that packet loss introduces random perturbations in the execution of ELA, allowing sensors to explore a vaster set of solutions and then retain the best one. We wonder if this improvement in performance would occur in a real WSN where packet loss is probably unevenly distributed. We note that, even though packet loss improves ELA accuracy in our simulations, it increases both convergence time and amount of necessary radio transmission.

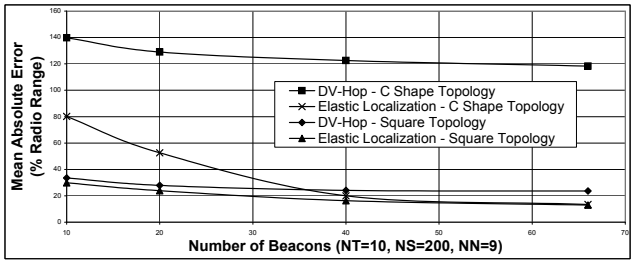

Fig. 24. Comparison of DV-Hop with ELA.

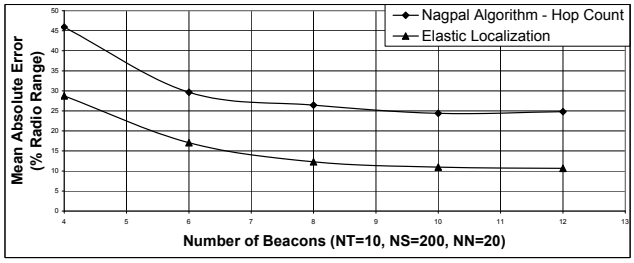

Fig. 25. Comparison of Nagpal's algorithm with ELA.

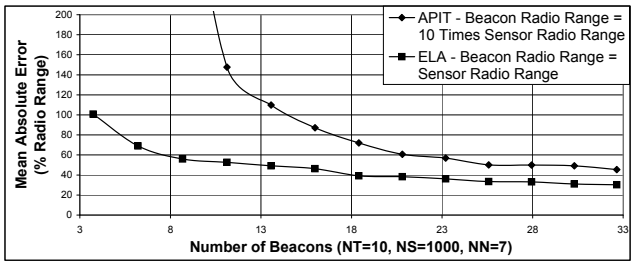

Fig. 26. Comparison of APIT with ELA.

E. Comparison with State of the Art Algorithms

In this section, we compare ELA with three state of the art algorithms: DV-Hop [12], Nagpal's solution [9], and APIT [11].We did not run simulations of the corresponding algorithms. Instead we directly use the results provided in the respective papers and compare with ELA using the same network size, average number of neighbors, and number of beacons.

Figure 24 compares ELA to DV-Hop. We use two topologies: a square topology (see Figure 17(a)) and a C topology (see Figure 17(d)). The iid randomly generated networks contain 200 sensors, sensors possess nine neighbors on average. ELA performs significantly better than DV-Hop on the square topology: the MAE on position estimation decreases of up to ten points. Besides, ELA obtains much better results with a C topology. In particular, when more than fourty sensors are beacons, the performance of ELA in a C topology is similar to its

performance on a square topology: ELA achieves a MAE of less than 20% whereas DV-Hop MAE has a value of more than 110%.

Figure 25 compares ELA to Nagpal's solution. We simulate iid randomly generated networks of 200 sensors. Sensors possess twenty neighbors on average. We observe a significant and constant improvement of fifteen points over Nagpal's algorithm.

Figure 26 compares ELA to APIT. The comparison is delicate as APIT uses beacons that have greater radio ranges than sensors. We use iid randomly generated networks of 1000 sensors where sensors possess seven neighbors on average. We give a significant advantage to the APIT network by considering that its beacons possess ten times the radio range of its sensors. The ELA beacons have the same radio range as their sensors. Even with this handicap, ELA performs better than APIT, particularly when the number of beacons is less than thirteen. For a greater number of beacons, ELA performance is still better than APIT performance by at least fifteen points.

Finally, we note that the communication overhead of ELA is probably higher than the one of DV-Hop, Nagpal's algorithm, and APIT. Indeed, sensors using ELA must broadcast information about themselves, their neighbors, and their b_{max} closest beacons. By contrast, DV-Hop and Nagpal's algorithms need only to propagate information about beacons. APIT, using beacons with high radio range, has relatively low communication needs. We nevertheless provided data demonstrating that ELA communication overhead do not impede its efficient implementation on a real WSN (see Section IV-A).

F. Mobility

Our algorithm can operate in the presence of mobile sensors if it undergoes the following modification. In the static version of ELA, sensors receive localization datagrams from their neighbors and subsequently update their tables, as described in Section III-A. In the mobile version of ELA, sensors additionally keep track of which neighbors send a given piece of information and when the information is received. Sensors decide that a given piece of information is outdated when $t_{current} - t_{info} > t_{limit}$, where $t_{current}$ is the current time, t_{info} is the time at which the information was received, and t_{limit} is the time after which a given piece of information is discarded if it has not been refreshed. For our simulations, we equal t_{limit} to the broadcasting period T.

As in [13], we adopt the random waypoint mobility model. Both beacons and sensors are mobile and initially iid randomly distributed within a $500m \times 500m$ square area. They iid randomly select a destination within the same square and start moving towards this destination with a velocity v iid randomly selected within the interval $[0, v_{max}]$ where v_{max} is the maximum velocity that sensors can reach. Once they reach their goal, sensors select another destination, another velocity and proceed immediately, without any pause. We note that when sensors constantly move, the ELA algorithm remains constantly in Phase I.

Figure 27 reports on the convergent behavior of ELA in mobile networks. We simulate networks of size 500 where mobile sensors are initially iid randomly positioned within a square area. The networks contain sixteen mobile beacons, the average number of neighbors in the mobile network is thirteen. The graph presents the evolution of the MAE on position estimation over time. Note that the time unit is equal to the broadcasting period T. We test the behavior of the networks for various maximum velocities varying from 0% to 160% of the radio range per time unit. Unsurprisingly, we observe that the slower the maximum

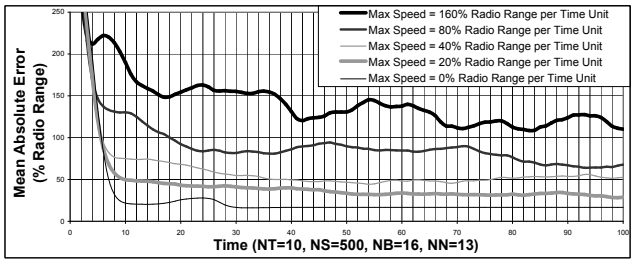

Fig. 27. Convergence of ELA in Mobile Networks.

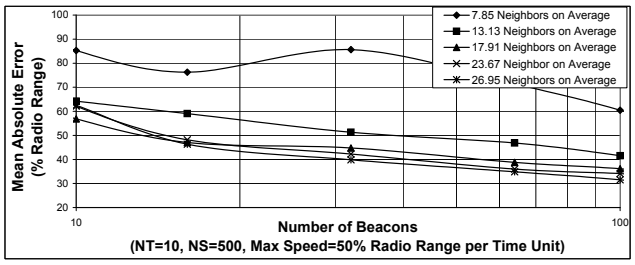

Fig. 28. Influence of Sensor Density and Number of Beacons on ELA Performance in Mobile Networks.

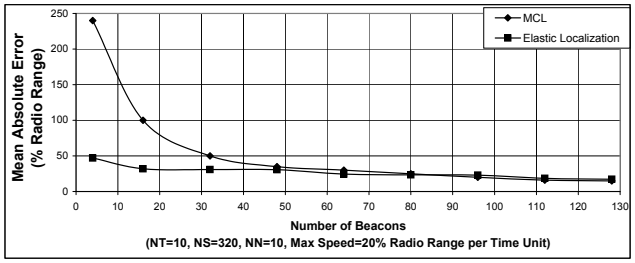

Fig. 29. Comparison of MCL with ELA.

velocity is, the better ELA performs. We note that for maximum velocities of up to 40% of the radio range per time unit, the MEA on position estimation remains below 50% of the radio range.

In Figure 28, we observe the variations of ELA performance in mobile networks according to the number of beacons and the average number of neighbors. The experimental setup is similar to the one of the previous experiment except that the maximum speed is 50% of the radio range per time unit. We observe that the MAE on position estimation steadily improves as the number of beacons increases and reaches 30% of the radio range in the best case. Low connectivity negatively impacts the results. In particular, we observe that an average number of neighbors less than thirteen is not suitable for mobile networks with sixteen beacons and a maximum speed of 55% of the radio range per time unit.

Figure 29 compares ELA to MCL [13], a localization algorithm for mobile sensor networks using Monte Carlo filtering techniques. We simulated networks containing 320 sensors and where the average number of neighbors is ten. As in previous experiments, the sensors are iid randomly distributed within a square area. We observe that ELA performs better than MCL for networks containing less than 15% of beacons. When the number of beacons exceeds 15%, MCL and ELA perform equally well. We note that the communication overhead of ELA is higher than the one of MCL. Indeed, MCL sensors only broadcast information about themselves and recently heard beacons while ELA sensors broadcast information about themselves, their one-hop neighbors, and their b_{max} closest beacons.

V. Conclusion and Future Work

ELA is a distributed, scalable, robust, efficient, and accurate algorithm. It only assumes that a few percent of the sensors know their real physical position and that an estimation of the maximum communication range is available. It operates with any kind of network topology and no time synchronization algorithm is necessary for its functioning. Additionally, ELA behaves remarkably well in presence of asymmetry, packet loss, and mobility. It also tolerates errors on maximum radio range and beacon position estimations. Furthermore, ELA performs better, in terms of localization error, than four state of the art algorithms in conducted experiments.

ELA still necessitates further study. Particulary, we are interested in implementing ELA on a real WSN platform. In addition, we are working on using received radio power and angle of arrival to further improve the performance of ELA. We also note that all our equations remain valid when the sensors are distributed within a three dimensional space and consider testing the performance of ELA in such a space. Finally, we attempt to adapt ELA to heterogeneous networks that use various sensors with different radio range capabilities.

References

[1] A. Harter and A. Hopper, "A distributed location system for the active office," *IEEE Network*, vol. 8, no. 1, 1994.

[2] "Ascension Technology Corporation," http://www.ascension-tech.com.

[3] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positions System: Theory and Practice*, fourth edition ed. Springer Verlag, 1997.

[4] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," in *Mobile Computing and Networking*, 1999, pp. 59–68. [Online]. Available: citeseer.ist.psu.edu/harter99anatomy.html

[5] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Mobile Computing and Networking*, 2000, pp. 32–43. [Online]. Available: citeseer.ist.psu.edu/priyantha00cricket.html

[6] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Mobile Computing and Networking*, 2001, pp. 166–179.

[7] K. Whitehouse, "The design of calamari: an ad-hoc localization system for sensor networks," Master's thesis, University of California at Berkeley, 2002.

[8] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *Proceedings of INFOCOM 2003*.

[9] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *the 2nd International Workshop on Information Processing in Sensor Networks (ISPN'03)*, April 2003.

[10] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices, technical report 00-729," April 2000.

[11] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM Press, 2003, pp. 81–95.

[12] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, January-April 2002.

[13] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM Press, 2004, pp. 45–57.

[14] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *Proceedings of National Telecommunications Conference*, 1978, pp. 4.3.1–4.3.5.

[15] "Mica2 and Mica2Dot platforms," http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.

[16] J. C. Navas and T. Imielinski, "Geocast - geographic addressing and routing," in *Mobile Computing and Networking*, 1997, pp. 66–76.

[17] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," 2002.