Local Positioning for Environmental Monitoring in Wireless Sensor and Actor Networks

Mustafa İlhan Akbaş*, Matthias R. Brust†, and Damla Turgut*
*School of Electrical Engineering and Computer Science
University of Central Florida, Orlando FL
Email: {miakbas,turgut}@eecs.ucf.edu
†Computer Science Division
Technological Institute of Aeronautics, Brazil
Email: matthias.brust@ita.br

Abstract—Location estimation of sensor nodes is an essential part of most applications for wireless sensor and actor networks (WSAN). The ambiguous location information often makes the collected data useless in these applications. Environmental monitoring in particular, relies on an accurate position estimation in order to process or evaluate the collected data. In this paper, we present a novel and scalable approach for positioning of mobile sensor nodes with the goal of monitoring the Amazon river. The actors in the scenario are stationary and positioned at reachable spots on the land alongside the river whereas sensor nodes are thrown into the river to collect data such as water temperature, depth and geographical features. The actors are not equipped with positioning adaptors and they are only aware of their distances from the other actors. The sensor nodes collect data and forward it to the actors. While floating in the river, sensor nodes are often multiple hops away from the actor nodes, which makes it challenging to apply traditional positioning techniques. Through extensive simulations, we show that the positioning of the nodes is feasible using a multi-hop approach with local information exchange only.

I. Introduction

Wireless sensor and actor networks (WSANs) are geographically distributed systems with large numbers of tiny sensor nodes and a limited number of actors [1]. The sensor nodes sense and observe the events in the environment and the more powerful actors collect and process information from the sensor nodes and react to the events. WSANs have a broad range of applications such as environmental monitoring, military surveillance and intelligent transportation. Besides the proper sensor adapters that detect movements, measure temperature or other environmental features, most applications require location information to be associated with the collected data in order to deliver meaningful results [2].

Positioning is defined as the process of estimating the positions of nodes in the environment. The positioning methods generally use *a priori* information about the environment such as the positions of some specific nodes or possible positions in the environment that the nodes can be located at [3]. Other sources for location estimation are the measurements made by the nodes and the communication among the nodes. The sensor nodes can gather various kinds of information when equipped with appropriate measurement technologies [4]. The measurements of distance between two nodes, network connectivity and direction or angle of signal arrival are widely used aids to support positioning in WSANs.

The environment considered in this work is a river going through dense Amazon rain forest (see Fig. 1). The actors are positioned at specific accessible spots on each side of the river, while the sensor nodes are thrown into the water. As the sensor nodes float, they collect information about the environment and transmit the collected data to the actors. The data are analyzed at the actors to map the characteristics of the river. The positioning process is critical in this environment to match the collected data with the position of a sensor node at a given time. In many scenarios, the collected data become unusable if not associated with the position and the time.

The river scenario introduces particular challenges such as a continuous change of the communication topology. Interference techniques and time stamps are used in order to overcome these challenges and to reveal the paths that the sensor nodes follow through the river. Hence the main goal of the positioning algorithm is not creating a self-awareness of locations at sensor nodes but enriching the collected data with positioning information.

The paths of the nodes deliver additional information about the nodes and the environment such as speed, direction or the structure of the terrain. For instance, the detection of a significant change in the speed of a majority of the nodes in a certain location of the river may indicate that either the terrain is more inclined in that area or there exists a small waterfall. Another example would be that if at a certain point of the river, the differences among the speeds of the nodes are large, some nodes may have run into obstacles while the others flow through the river without any problems.

The main contribution of this work is the design of an algorithm to enrich the collected data with localization information even if the sensor nodes are multiple hops away from the actor nodes. For this purpose, the sensor nodes create a hierarchical network structure. The network organization is used along with lateration for localization and it is dynamically adapted for energy efficiency. The accuracy of the event localization is improved by taking the estimation of the paths followed by the nodes into consideration.

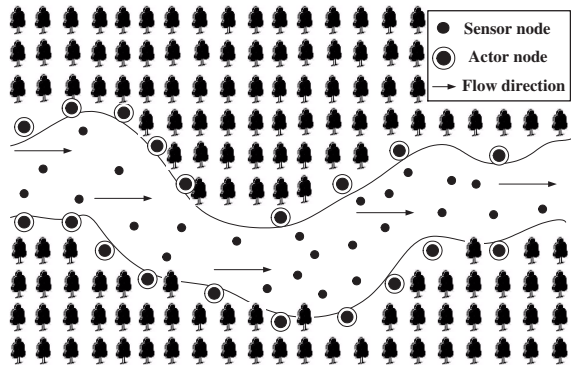The remainder of this paper is organized as follows. Re-

Fig. 1. The sensor nodes are floating in a river going through the dense Amazon rain forest and the actors reside on the accessible spots.

lated work is given in Section II. The proposed algorithm is presented in Section III. We demonstrate the validity and applicability of our method with simulations in Section IV and finally conclude in Section V.

II. RELATED WORK

Positioning in WSNs and WSANs has been attracting significant interest in the last decade due to the realization of low-cost and multifunctional sensor nodes and their deployments in both indoor and outdoor environments [5]. Although the positioning is significantly important in most of the deployed systems, implementation of algorithms is challenging since the positioning techniques in WSANs have their particular constraints. For instance the deployment areas are generally complex and they have issues such as accessibility, line-of-sight, and so on. The measurement method for the positioning algorithm also needs to be selected according to the environment, the limited capabilities of the sensor nodes and the conditions of the scenario. The heterogeneous node structure of WSANs is yet another challenge and requires a clear differentiation in the roles of the nodes in the network.

The proposed algorithms in this field make use of various approaches such as graph theory [6], multidimensional scaling [7], distance vectors (DV) [8], particle filters (e.g. [9], [10], [11]), recursive systems [12] and genetic algorithms [13]. These methods utilize diverse measurement techniques, the main types of which are time of arrival (ToA) [14], angle of arrival (AoA) [15], [16], received signal strength (RSS) [17], and distance related measurements.

We adopt a multi-hop positioning approach for our algorithm. There are many well known multi-hop algorithms such as the distributed hop-by-hop positioning algorithm, APS, proposed by Niculescu and Nath [8]. APS works as an extension of distance vector routing and GPS positioning to provide the approximate locations for all nodes in a network where only a limited fraction of nodes have self location capabilities. An alternative approach by Savvides et al. [18] uses the sensor nodes, after their positions are estimated, as the anchor nodes in a multilateration algorithm. This process is called iterative multilateration, which is also employed by Savarese

et al. [19]. The critical observation in these algorithms, which is also true for our algorithm, is that at least three actor nodes are needed to determine the position of a sensor node. Nagpal et al. [20] organizes a global coordination system in the network by estimating the Euclidian distance of a hop. The estimation algorithm uses the number of communication hops of the sensor nodes and the position error is minimized with imperfect distance estimates. The important theoretical results given by Nagpal et al. are the critical minimum average neighborhood size for sufficient accuracy and the limit on the resolution of a coordinate system determined by local communication.

In contrast to most of the discussed approaches, the positioning information of actor nodes do not need to be forwarded in our algorithm. In other words, since providing self-awareness for the sensor nodes is not an objective of the algorithm, the sensor nodes are not informed of their position information. The particular scenario discussed in the paper has an additional distinct and important constraint, which is the high mobility of the sensor nodes.

The organization of the overlay network scheme for assignment of the hop-counts and affiliation with the actor nodes is a critical part of our algorithm, where clustering is employed as the solution. The actor nodes function as the permanent clusterheads and the structure allows multi-hop clusters, which is different than most of the existing clustering algorithms for mobile ad hoc networks (eg. [21], [22], [23], [24]). KHOPCA [25] is a multi-hop clustering scheme consisting of a set of simple and easy-to-implement locally acting rules, which has an approach similar to our algorithm, but it does not fully satisfy the specific requirements of our scenario such as predetermined clusterheads.

III. MULTI-HOP POSITION ESTIMATION

The proposed multi-hop position estimation is described in detail in this section. It is important to note that the main goal of the approach when estimating the locations of the sensor nodes is to enhance the information collected from the river. The system is not intended to detect a specific event (e.g. fire or an intruder); instead it is used to monitor the river for a period of time and to create a detailed set of data for a complete analysis of the river. Therefore, different than most of the existing localization algorithms, the sensor nodes are not informed about their positions and no computation at the sensor nodes is specifically required by the positioning algorithm.

A. Network layout

The network, N, consists of a set of stationary actors A, a set of mobile sensor nodes S and a sink node, which is in charge of the data aggregation. The sensor nodes and the actors have transmission ranges r_s and r_a, where $r_a > r_s$. Obtaining location information by recording positions manually or through GPS are not viable options for sensor nodes due to their limited resources. Additionally, thick forest structure may block satellite communication from time to time as the

sensor nodes float in the river. On the other hand, the actors are placed at predetermined accessible spots in the area. Therefore only actors have a priori knowledge of their positions and each sensor node can directly communicate only with its immediate neighboring nodes.

The actors acquire additional knowledge about their positions either from an external source or the position information is encoded as they are placed in the area. The actors are positioned such that each one has at least one actor or sink in its transmission range. Each actor uses its full transmission range to communicate with the other actors and the sink. The network among the actors and the sink is defined as the backbone of the network. The backbone is formed by selectively flooding a packet originated at sink among actors. Thus, each actor node learns the neighboring actor, which is set as the destination for packets forwarded to the sink.

B. Weighing the nodes

There is no initial configuration on the sensor nodes to affiliate them with the actors. Each sensor node keeps only a "weight" value for each actor it is affiliated with, which represents the minimum number of hops required to reach the actor from that sensor node. The sensor node keeps the list of the weights for its affiliated actors in a "weight table".

The sensor nodes initially take random weight values between 0 and $k-1$ while each actor node is assigned to a constant weight k. The only data available for a sensor node s as it floats in the river are the direct neighbors $Neigh(s)$ and their corresponding weights $w(Neigh(s_i))$. Hence, the sensor nodes maintain and update only local information.

When communicating with the sensor nodes and creating the affiliation areas, the actors use the same transmission range, r_s, as the sensor nodes and exchange packets only with their neighboring sensor nodes. Each actor node encodes the packets transmitted with its ID and weight k. These values demonstrate that this packet is coming from an actor. Each sensor node updates the weight values in its records via local updates. Thus the information about each actor is distributed to the affiliation area of that actor.

A sensor node keeps the maximum weight for each actor it receives packets from, as depicted in Figure 2. Hence, a sensor node is capable of being affiliated with multiple actors. When a sensor node receives a packet from an actor, the packet is retransmitted if the weight value on it is less than the sensor nodes weight for that actor. Otherwise the sensor node drops the packet to avoid unnecessary traffic and energy consumption in the network.

The condition, in which a sensor node doesn't receive any weight updates, is defined as the "loss of connection" for the sensor node. In that case, the sensor node sets its hop value to the minimum value defined for the network. Then it operates only in "listening" mode and does not transmit any packets.

Algorithm 1 describes the state transitions for the node v depending on its actor affiliation. The weight w of each affiliation corresponds to "k-hop distance" of a node v to an actor a.
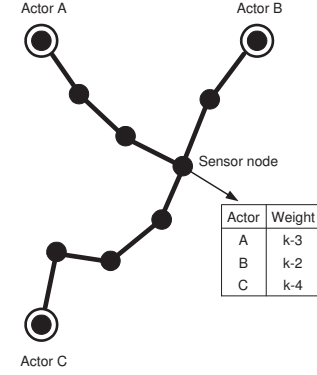

Fig. 2.   One sensor node affiliated with multiple actors. The sensor is maintaining a list of actor affiliation and weight.

Algorithm 1 The state transitions of a node v

1: $w_a(v)$: The weight of node v for actor a
2: $\max(w(Neigh(v_a)))$: M
3: **if** v is a sensor node **then**
4:     **if** v is not affiliated with an actor node **then**
5:         $w_a(v) = 0$
6:     **else if** $M = k$ **then**
7:         $w_a(v) = k - 1$
8:     **else if** $M! = k$ & $M > w_a(v)$ **then**
9:         $w_a(v) = M - 1$
10:     **else if** $M < w_a(v)$ **then**
11:         $w_a(v) = w_a(v) - 1$
12:     **else if** $Neigh(v_a) = $ Empty **then**
13:         $w_a(v) = 0$
14:     **end if**
15: **else**
16:     $w_a(v) = k$
17: **end if**

The hierarchical structure of the network is created with the step, by which a sensor node continuously adapts its weight to its local neighborhood. For each of its affiliated actor node, a sensor node changes its weight to "the highest neighbor weight – 1" unless it already has the same weight value. Hereby a hierarchical structure is dynamically formed and updated.

The next step covers situations where higher weight nodes, which are not actors and not affiliated with any actors, attract surrounding nodes with lower weights. According to this step, a higher weight node successively reduces its weight, finally connecting to an existing actor or until its weight becomes zero. In a case where node v is not affiliated to a particular actor, its weight is set to zero.

Figure 3 shows an example of a sensor node's weight adaptation according to the algorithm described above. All the other nodes in this example are assumed to be stationary and k value for the network is set to four. The mobile node in the figure is initially not affiliated to any actors, so its weight is zero. Then the node becomes directly connected to an actor and its weight becomes $k - 1$. As the node moves away from
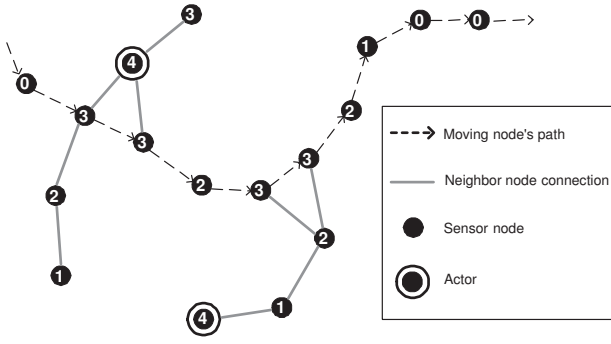
Fig. 3. Weight adaptation of a sensor node

Fig. 4. The weight tables of sensor nodes affiliated with two actors.

the actor node and affiliates with another actor node, its weight changes according to "the highest neighbor weight – 1" rule. Finally, it ends up not being affiliated with any actor node, its weight is decreased until it becomes zero and stays there as stated in the last step.

C. Adapting k

One of the characteristics of our algorithm is that the nodes use only local information. The sensor nodes update their data locally as the network topology changes due to node mobility. The sensor nodes affiliated with an actor form the "affiliation area" of that actor. In other words, if the neighbors of a sensor node which are affiliated with an actor, have zero weights, then that sensor node cannot affiliate with that actor and it is out of the coverage area.

When a sensor node is outside of all the affiliation areas, it is called an "unconnected node". Unconnected nodes with weight zero are physically able to exchange packets with neighbors that have zero weights, but there is no communication among these nodes. Therefore the selection of the k value for the actors is critical for the traffic and the energy consumption in the network. In our algorithm, actor nodes adapt their k values according to their observations. In order to find k, each actor follows the following steps:

- The k is initialized with a high value (depending on the size of the network area) at the actors. If the k value of an actor changes, the actor announces this change to the other actors via the backbone.
- As the sensor nodes float into the reception range of actor nodes, they start to form and update their weight tables as shown in Figure 4. Actors also keep weight values for the other actors with the same method.
- Each actor computes a k_u value for all of its neighbor actors as follows:

$$k_u = \frac{k_n - w_n}{2}$$

where k_n is the neighbor's k value and w_n is the weight value of the actor for that neighbor actor. k_u represents the minimum number of hops between two actors. The k_u values are updated continuously such that the minimum is kept in the records for each neighbor actor.
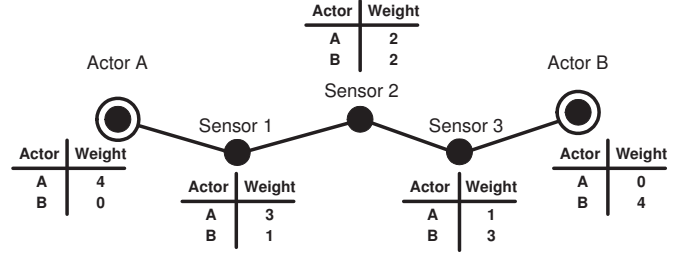
- The distances from the neighbor actors are calculated. Actors have their location information, so the distance (d) between two actors, $i = 1, 2$, is calculated simply by using the Pythagoras theorem:

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = d^2$$

- The actor's k value is updated with the maximum of the k_u values.

There are several aspects of k-adaptation which are critical for the algorithm. First, an actor's affiliation area is limited by adapting k, which reduces the number of packets that sensor nodes transmit as they update their weights since they have fewer actors to keep weights for. As the packet transmission drops, the energy consumption also decreases in the network. The other important aspect of k-adaptation is the hop-distance estimation since the values calculated with this method are used when estimating the locations of the sensor nodes. Moreover, many multi-hop localization algorithms in the literature become ineffective as the topology becomes irregular. Our algorithm's adaptation of k forces sensor nodes to be affiliated only with actors in close proximity which reduces the negative impacts of the irregular topology.

D. Positioning the nodes

Each actor in the network estimates the 1-hop distance (k_u) for each actor it keeps a weight value for. The method described in the previous section is used by the actors for this purpose. The average of the calculated k_u values are used as the 1-hop distance value. Then the average 1-hop-distance, estimated as $\frac{d}{2k_u}$, is used for localization purposes.

Since the sensor nodes dynamically keep and update their weights, the hop-distances of the sensor nodes to the actor nodes are known in the network. The sensor nodes piggyback their weight tables to the data packets to inform the actor nodes. These weight tables are used to calculate the location estimation for nodes. The estimated distance of a sensor node to an actor A is calculated as follows:

$$d_A = w_A * h_A$$

where d_A is the estimated distance to the actor A, w_A is the weight of the sensor node for the actor A and h_A is the average 1-hop distance of actor A.

The distance estimation is based on the number of hops needed to reach from the sensor nodes to the actor ndoes. Therefore it's similar to the distance vector (DV) based approach of Niculescu and Nath's model [8]. However, Niculescu and Nath's model is designed for static networks and additionally, in most of the DV-based solutions, sensor nodes need to assign a fixed memory to save the locations of all the landmarks, hop-counts to these landmarks and average hop-distance values. In our approach, a sensor node keeps only the weight values for its affiliated actor nodes. All the other computation and memory requirements of the algorithm are handled by the actors and the sink, which is a better-fit for WSAN structure in terms of the usage of memory, computational resources and energy. Hereby, the information flooding, which is common and intense in DV-based solutions, is also minimized.

When the sensing sensor node transmits its weight values for three actors, its distances to these actors are estimated according to the "1-hop distance" value and the weights. Then, its position is calculated based on the acquired distances and the localization information obtained by the lateration.

The estimated distance of the sensor node is calculated for each of its affiliated actor and then these values are plugged into the lateration operation for the estimation of the sensor nodes position. When a sensor node is affiliated with multiple actor nodes, its position estimation can be represented with a system of equations, written in matrix form as follows:

$$
\begin{bmatrix}
2(x_n - x_1) & 2(y_n - y_1) \\
\vdots & \vdots \\
2(x_n - x_{n-1}) & 2(y_n - y_{n-1})
\end{bmatrix}
\begin{bmatrix}
x_s \\
y_s
\end{bmatrix} =
$$

$$
\begin{bmatrix}
(d_1^2 - d_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\
\vdots \\
(d_{n-1}^2 - d_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2)
\end{bmatrix}
$$

where $(x_i),y_i)$, $i = 1,...n$, are the positions of the n actors and d_i, $i = 1,...n$, are the estimated distance values. The solution pair of this system, (x_s,y_s), which minimizes $\|Ax - B\|_2$, is the pair that minimizes the mean square error, where $0.5A$ is the left hand side matrix, **B** is the right hand side matrix and **x** is the vector for (x_s,y_s) pair. Since $\|.\|_2$ is minimized, the system is solved with minimum average error for all positions of the actors that the sensor is affiliated with.

For any vector v, $\|v\|_2^2$ is equal to $v^T v$, which can be used to find a solution for **x**. Therefore, if the same expression is written for $\|Ax - B\|$, an equation for **x** can be found:

$$\|Ax - B\|_2^2 = x^T A^T Ax - 2x^T A^T B + B^T B$$

This expression is minimized when the mean square error is minimized. Therefore, the gradient of the expression has to be set to zero considering it as a function of **x**:

$$2A^T Ax - 2A^T B = 0 \Leftrightarrow A^T Ax = A^T B$$

This is the normal equation for the linear least squares problem and it has a unique solution. We use the Cholesky

TABLE I
SIMULATION PARAMETERS

number of sensor nodes	1-25
number of actor nodes	25-50
k value	1-5
total area	250x250 meters
sensor node transmission range	40 meters
sensor node floating speed	1,25-1,5 m/s

factorization to solve this equation and to get the estimations for x and y coordinates of the sensor nodes. Accordingly, the resulting matrix is given by:

$$x = A^T (AA^T)^{-1} B$$

The estimations for the coordinates of the sensor nodes are calculated by this method for each time instance and recorded. Since our goal is not an online location-awareness for the sensor nodes, the path the nodes followed is estimated after all of these points are calculated. Essentially, the paths are estimated using the interpolation of these coordinates. Hence, the effect of the errors in the individual estimates to the paths of the sensor node are minimized.

IV. SIMULATION STUDY

In this section, we evaluate our approach by measuring how the estimated location errors vary with different network and algorithm characteristics. The transmission range of a sensor node is set to 40 meters, a realistic range for a sensor node (Cerpa et al. [25]). The number of the nodes and the value of k are used as the simulation parameters in the experimental setup. The communication graph is built according to the system properties specified in Section 3. The actor nodes are stationary, deployed in the area uniformly at random with the constraint that they are able to form a connected graph for communication among themselves. Sensor nodes are mobile and they are flowing from left to right in the watercourse. The communication links among the sensor nodes are open to failure. Table I summarizes the simulation parameters used in our simulation study.

A. Experiment 1

1) 1-hop clustering: In order to investigate the affect of clustering on the performance of the proposed algorithm, a scenario with only a single sensor node and 25 actor nodes is considered. The main purpose of this experiment is to observe the performance difference between 1-hop clustering and multi-hop clustering. In this scenario, a single sensor node floods from left to right with a speed between 1.25 to 1.5 m/s and connects to the actor nodes whenever it is in the transmission range. Hence the sensor node can be connected to multiple actors and the parameter k has no influence on the location estimation in this experiment.

Figures 5 and 6 show the exact trajectory of the node on the x and y coordinates respectively. The x-axis of the figures represents the time steps. The node is moving from 0 to 250m in x coordinate and for y coordinate the node stays
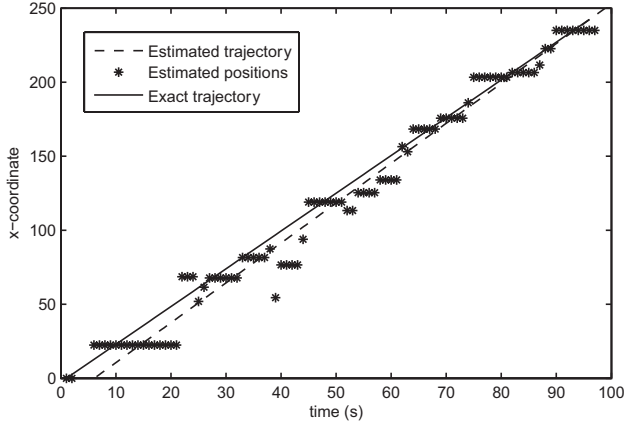
Fig. 5. Estimated x-coordinates and the real trajectory of a sensor node for the experiment 1 with 1 sensor node and k=1.

Fig. 7. Estimated x-coordinates and the real trajectory of a sensor node for the experiment 1 with k=3

on the position around 110m. The network density is high enough that a position could be calculated for each time step. In other words, the node was affiliated with four actors for most of the time. In Figures 5 and 6, the estimated trajectory has been produced using linear regression over the data set of the estimated positions. The estimated positions are with an error of 30 meters according to the exact trajectory, but most estimated positions show much less error. The error is as expected since the transmission range is 40 meters and the estimation is based on the estimated hop-distance. However, the estimated trajectory of the considered sensor nodes is within a range of at most 10 meters, but mainly within 5 meters or less.

2) k-hop clustering: The settings in the 1-hop clustering experiment have been modified such that 25 actors and 25 sensor nodes are participating in the simulation area and the parameter k is set to three. In this case, the maximum length from a sensor node to an actor node can be three hops. In similar illustrations as in Figures 5 and 6, the accuracy of the
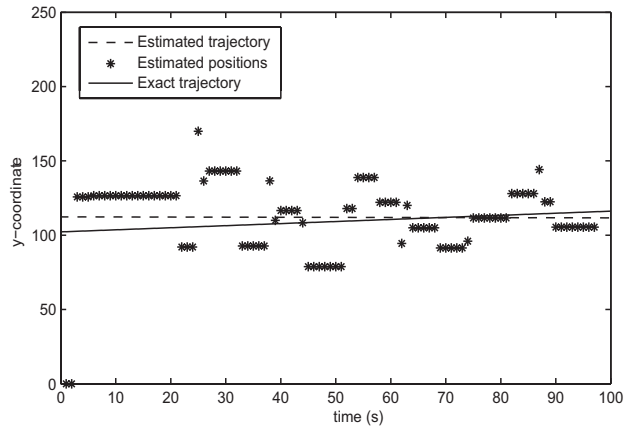

Fig. 8. Estimated y-coordinates and the real trajectory of a sensor noder for the experiment 1 with k=3

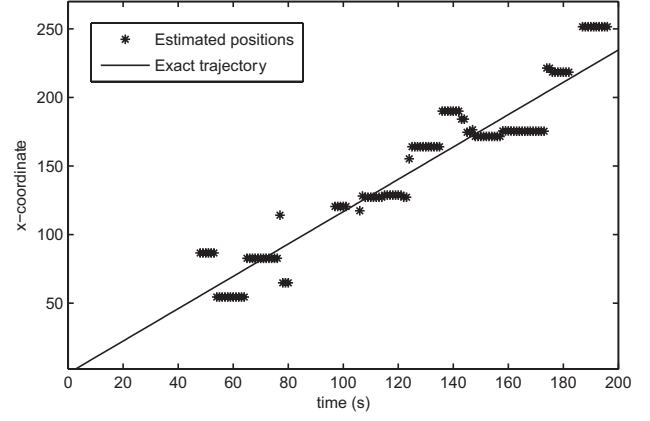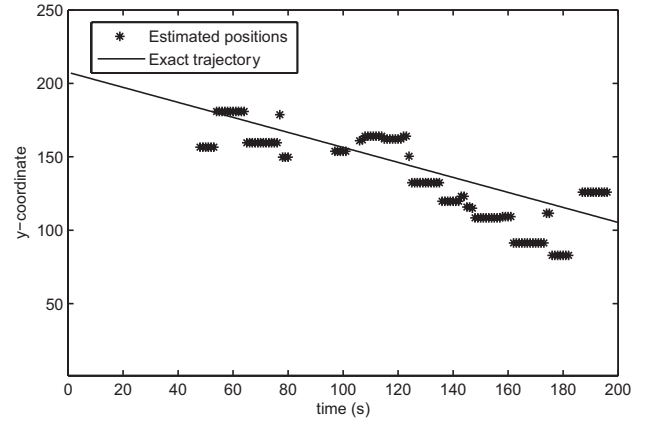location estimation of one of these 25 nodes under the use of local k-hop clustering information is shown in Figures 7 and 8. The sensor node is connected to the actor networks via intermediary nodes, where k=3, for most of the time in this experiment. Therefore multi-hop information is used. The applied algorithm estimated the hop-distances using far less accurate data. However the results show that the estimated positions are very close to the exact trajectory. Compared to the previous experiment, the error is only slightly higher. The main difference is that there are more time steps where there is not enough information that could be collected to estimate the position. It should be noted that this result also depends on the real trajectory of the individual sensor node.

B. Experiment 2

In this scenario, the number of nodes is 50, 25 of which are the actors and 25 are the sensor nodes. However, in order to get the accuracy for single hop settings, the parameter k is set to 1. The error distribution is illustrated in Figure 9. The error has a normal distribution and it ranges from -30 to
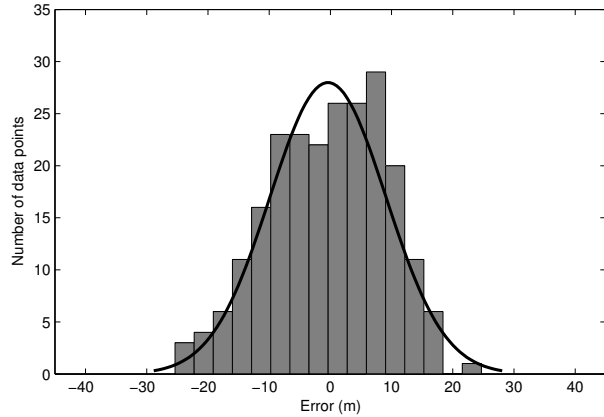

Fig. 6. Estimated y-coordinates and the real trajectory of a sensor node for the experiment 1 with 1 sensor node and k=1.

Fig. 9. The error distribution of the estimation results for the experiment 2 with k=1
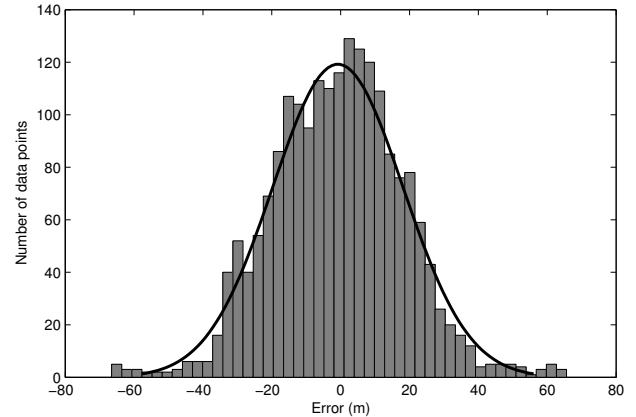

Fig. 10. The error distribution of the estimation results for the experiment 3 with k=3

Fig. 11. The error distribution of the estimation results for the experiment 4 with k=4

30 meters. Hence this error is for the x-coordinate estimation. Since the nodes are floating almost horizontally, the accuracy on the y-coordinate is relatively good. We can also observe that the number of estimated points is not as high as the previous experiment.

C. Experiment 3

The multi-hop clustering approach for location estimation is a core part of the contribution of this paper. In this experiment, in order to investigate the usage of multi-hop clustering, the number of sensor nodes is increased while the number of actor nodes remains equal. Thus actors that are not in the transmission range of a single mutual sensor node must make use of the multi-hop information provided by the passing sensor nodes. The k parameter is set to three allowing the nodes to connect to an actor through another sensor node. The resulting Figure 10 shows that the error distribution is similar to the Figure 9 where the sensor nodes were connected directly to the actor nodes. The error rate is again normal distribution. However, there are two differences: (i) more positions could be estimated which means that the local multi-hop approach applied in our algorithm is able to gather more localization estimation than it does with a traditional approach, and (ii) the estimation accuracy is decreasing. Compared to the experiment 2, where the distribution was in between -30 and 30 meters, the distribution is now in between -60 and 60 meters.

We observe that higher the value of k is, more positions can be estimated by our algorithm. However, the accuracy decreases at the same time with the increasing k. Therefore the selection of k is an important issue for the network, which depends on the number of nodes, the environmental conditions and the application requirements.

D. Experiment 4

In this experiment, the settings remained the same as the experiment 3 except for the value of k. The value of k is increased to five, so clusters with more hops compared to the experiment 3 are allowed in the network. The error distribution
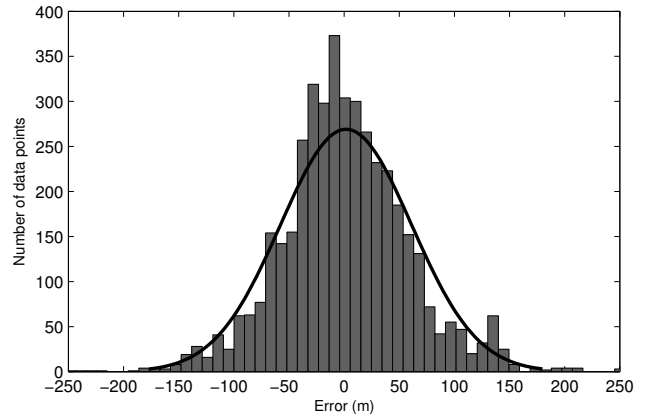
of the experiment is given in Figure 11. Similarly, Figure 11 shows that as the value of k becomes higher, the accuracy decreases further while the number of estimations increases due to further connection chains in the network.

V. CONCLUSION

In this paper, a novel and scalable algorithm for multi-hop localization of mobile sensor nodes is proposed. The goal of the proposed algorithm is to improve the on-site monitoring of a river in dense Amazon forest structure with a scalable approach. The particular scenario has its own challenges, which require robustness and high adaptability to failures and/or additional sensors. In order to overcome these challenges, the proposed approach does not create a self awareness of locations at sensor nodes since all communication in the system is locality preserving. The hop-counts, sensor node IDs and sequence numbers are used as the information related to localization. The algorithm is kept simple considering the limited lifetime of the WSANs, but it is possible to integrate an additional feature such as RSS to increase the accuracy of the results. Through extensive simulations, we have shown that the

localization estimation can be realized using local multi-hop information. In overall, as the multi-hop chains are allowed to become longer, more positions can be estimated with the cost of the accuracy. The selection of the maximum hop number is therefore an issue depending on the requirements of network.

As for the future work, we can employ the proposed algorithm to animal monitoring application. We also intend to use the localization information gained by the algorithm for data aggregation and dissemination within the network.

REFERENCES

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, October 2004.

[2] N. Patwari, J. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," in *IEEE Signal Processing Magazine*, vol. 22, no. 4, 2005, pp. 54–69.

[3] B. Turgut and R. P. Martin, "Using a-priori information to improve the accuracy of indoor dynamic localization," in *Proceedings of 12th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, October 2009, pp. 394–404.

[4] F. Regan, A. Lawlor, B. Flynn, J. Torres, R. Martinez-Catala, C. O'Mathuna, and J. Wallace, "A demonstration of wireless sensing for long term monitoring of water quality," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, December 2009, pp. 819–825.

[5] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[6] B. D. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang, "Graphical properties of easily localizable sensor networks," *Wireless Networks*, vol. 15, no. 2, pp. 177–191, 2009.

[7] H. Shi, X. Li, Y. Shang, and D. Ma, "Error analysis of quantised rssi based sensor network localisation," *International Journal of Wireless and Mobile Computing*, vol. 4, no. 1, pp. 31–40, 2010.

[8] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Journal of Telecommunication Systems*, vol. 22, no. 8, pp. 267–280, 2003.

[9] B. Turgut and R. P. Martin, "A multi-hypothesis particle filter for indoor dynamic localization," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, October 2009, pp. 742–749.

[10] L. Zhang, X. Zhou, and Q. Cheng, "Landscape-3D; a robust localization scheme for sensor networks over complex 3D terrains," in *Proceedings of IEEE Conference on Local Computer Networks (LCN)*, November 2006, pp. 239–246.

[11] B. Turgut and R. P. Martin, "Restarting particle filters: an approach to improve the performance of dynamic indoor localization," in *Proceedings of the IEEE GLOBECOM*, November-December 2009, pp. 1–6.

[12] H. Oliveira, A. Boukerche, E. Nakamura, and A. Loureiro, "An efficient directed localization recursion protocol for wireless sensor networks," *IEEE Transactions on Computers*, vol. 58, no. 5, pp. 677–691, 2009.

[13] J. Taheri and A. Zomaya, "A genetic algorithm for finding optimal location area configurations for mobility management," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, November 2005, pp. 560–567.

[14] I. Getting, "The global positioning system," *IEEE Spectrum*, vol. 30, no. 12, pp. 36–47, December 1993.

[15] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of Globecom*, November 2001, pp. 2926–2931.

[16] E. Elnahrawy, J.-A. Francisco, and R. P. Martin, "Bayesian localization in wireless networks using angle of arrival," in *Proceedings of ACM SenSys*, November 2005, pp. 272–273.

[17] B. Turgut and P. Martin, "Localization for indoor wireless networks using minimum intersection areas of iso-RSS lines," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, October 2007, pp. 962–972.

[18] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc wireless sensor networks," in *Proceedings of Annual International Conference on Mobile Computing and Networking*, July 2001, pp. 166–179.

[19] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of the Annual USENIX Technical Conference*, June 2002, pp. 317–327.

[20] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proceedings of the International Workshop on Information Processing in Sensor Networks*, 2003, pp. 333–348.

[21] M. Chatterjee, S. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, no. 2, pp. 193–204, April 2002.

[22] R. Marin-Perianu, H. Scholten, P. Havinga, and P. Hartel, "Energy-efficient cluster-based service discovery in wireless sensor networks," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, November 2006, pp. 931–938.

[23] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "TTDD: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 161–175, January 2005.

[24] A. Boukerche, R. Pazzi, and R. Araujo, "Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, November 2005, pp. 560–567.

[25] M. R. Brust, H. Frey, and S. Rothkugel, "Adaptive multi-hop clustering in mobile networks," in *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*, September 2007, pp. 132–138.