A Joint 3D Localization and Synchronization Solution for Wireless Sensor Networks Using UAV

Leandro A. Villas[†], Azzedine Boukerche[*], Daniel L. Guidoni[*‡], Guilherme Maia[*§] , Antonio A.F. Loureiro[§]
[*]PARADISE Research Laboratory - University of Ottawa - Canada
[‡]Federal University of São João del-Rei - Brazil
[§]Federal University of Minas Gerais - Brazil
[†]University of Campinas - Brazil
leandro@ic.unicamp.br, boukerch@site.uottawa.com, {guidoni, jgmm, loureiro}@dcc.ufmg.br

***Abstract*—Localization and synchronization are fundamental services in Wireless Sensor Networks (WSNs), since it is often required to know the position and the global time of sensor nodes to relate a given event detection to a specific location and time. However, the localization and synchronization tasks are often performed after the sensor nodes' deployment. Since manual configuration of sensor nodes is an impractical activity, it is necessary to rely on specialized algorithms to solve the localization and synchronization problems. With this in mind, in this work we propose a joint solution for the 3D localization and time synchronization in WSNs using an unmanned aerial vehicle (UAV). A UAV equipped with a GPS flies over the sensor field area broadcasting its geographical position. Therefore, sensor nodes are able to estimate their own geographical position and global time without the need of equipping them with a GPS device. By means of simulations, we show that our proposed joint solution leads to smaller time-synchronization and localization errors when compared to existing solutions.**

I. Introduction

A Wireless Sensor Network (WSN) can be defined as a cooperative network composed by thousands of small and resource-constrained sensor nodes [1]. The main tasks of a WSN are: monitoring physical phenomena and transmitting the collected data to a monitoring node called as sink. In this case, the sensor network is guided by events that usually generate data, which is forwarded to the sink. For a specific event to have meaning, however, it must be correlated in space and time in order to localize the event in the sensor field as well as the time that the specific event occurred. This correlation is performed using localization and synchronization systems.

Besides the task of localizing the collected data, a number of routing algorithms also uses localization information to improve their performance by creating routes that consider the nodes' position [3]. Moreover, synchronization systems can also be used to increase the performance of routing protocols. There are a number of routing algorithms that consider a transmission delay/schedule in order to increase the routing performance [4]. Finally, some algorithms consider a joint localization and synchronization solution in their design [3].

Typically, the solution for both systems use a recursive approach, such as [6], [7], where a node (unknown node) estimates its localization and clock time based on positions and clock times received from other nodes (reference nodes). When a node becomes a reference node, i.e., localized in space and time, it broadcasts its information to assist other nodes in their estimation. However, these solutions have some drawbacks such as error propagation due to estimation errors. Furthermore, in a 3D scenario, a node must receive at least four positions from reference nodes to estimate its own position, which may limit the number of nodes that are able to estimate their own position. Finally, to start the recursion process, $4-10\%$ of the network nodes must be equipped with a GPS receiver (beacon nodes), which increases the network cost.

This work aims to eliminate some of the drawbacks described above (beacon nodes and error propagation) of existing localization and synchronization approaches. In this paper, we propose a joint solution for the 3D localization and time synchronization systems that uses an Unmanned Aerial Vehicle (UAV) in wireless sensor networks. The UAV is equipped with a GPS receiver and traverses the sensor field broadcasting its position and clock time, thus allowing sensor nodes to estimate their position and clock. The proposed solution exhibits three main contributions for the localization and synchronization systems: (i) all network nodes are able to estimate their localization and local time with high accuracy; (ii) the proposed solution is efficient for both sparse and dense networks and (iii) the proposed solution reduces drastically the network cost.

The remainder of this paper is structured as follows. In the next section, we present the problem definition and in Section II we provide an overview of the existing approaches for localization and synchronization for WSNs. Our proposed solution is outlined in Section III, while the detailed performance evaluation and simulation results are analyzed in Section IV. Finally, in Section V we summarize our conclusions and future work.

II. Related Work

The related work is organized as follows. First, we present the literature localization algorithms and then, we present the synchronization solutions.

A. Localization Algorithms

Most of the literature algorithms improve the Ad Hoc Position System (APS) [8] or the Recursive Position Estimation

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

(RPE) [6]. In the APS, a reduced number of beacon nodes (at least 3) is deployed in the network. Each beacon node starts a broadcast message containing its position and each sensor node calculates the distance using multi-hop communication from each beacon. Once the distances are calculated, the sensor nodes can estimate their position using, for example, trilateration. The RPE algorithm uses a different approach. The sensor nodes estimate their position based on a set of beacon nodes. The algorithm is divided into four phases. In the first phase, each beacon node sends its position to its neighbors. In the second phase, when a sensor node receives the beacons' messages, it estimates the distance from each beacon using the RSSI technique. In the third phase, the unknown nodes estimate their position based on the received information. In the fourth phase, the unknown nodes become reference nodes and send their position to their neighbors, increasing the number of available position to be used to convert an unknown node into a reference node. The disadvantage of this algorithm is that the error in the position estimation is spread over the network, thus increasing the estimation error. There are other solutions that evolve from the APS and RPE algorithms by focusing on specific features in specific scenarios [9].

B. Synchronization Algorithms

The definition of time synchronization can be divided into three cases: (i) relative time synchronization, which is used to order messages and events; (ii) independent clock, where a node keeps track of drift and offset and (iii) global time synchronization, where there is a global time throughout the network. In this paper we are interested in the latter case. There are a number of synchronization algorithms to solve the global time synchronization in WSNs [10], [11]. In the state-of-the-art clock synchronization algorithm FTSP (Flooding Time Synchronization Protocol) [12], a node synchronizes its clock based only on a single message. FTSP takes advantage of MAC-layer time to send a message, called One Hop Synchronization (OHS). A root node, which has a synchronized global clock, creates a message with its clock and broadcasts this message to its neighbors. When an unsynchronized node receives this message, it gets the timestamp inside the message and adds to this timestamp a pre-defined OHS value and then, synchronizes its clock. FTSP was evaluated in a real wireless sensor network and the OHS presented a precision of 2–4 μs in a Berkeley Mica2 platform.

III. Proposed Solution

This section presents the proposed solution to the 3D localization and synchronization problems using Unmanned Aerial Vehicle (UAV). Our solution aims to integrate both problems, since they are related.

A. Localization and Synchronization Systems

The localization system can be divided into two phases: distance estimation and position computation. First we show how our solution estimates the distance between two nodes and then we show how it computes the nodes' position.

There are several methods to estimate the distance between two nodes [13]. The most commonly used method is the RSSI, since it requires no extra hardware besides a radio transmitter/receiver built into the sensor node. To calculate its 3D position, the sensor node needs at least four reference points. These reference points are provided by the UAV during its flight over the sensor field. When the sensor node has at least four reference points and the distance to each point, it is able to estimate its position. Multilateration is the most common method used to estimate the position when in possession of four or more reference points. In this work the least squares [14] method was used to solve the system of equation containing positions and distances.

To solve the synchronization problem we used the Flooding Time Synchronization Protocol (FTSP), where the network nodes synchronize their clock using one-way communication. To synchronize a clock with one-way communication, a node should calculate the sender time, MAC access time, propagation time and receiver time. The most important factor is the MAC access time. The sender time is the time to create a message to transmit on the network and the receiver time is the time to receive a message and transmit to the host. This time can be softened if the timestamp is attached to the message in the MAC layer, just before transmitting. The propagation time can be easily calculate for a given propagation model. Finally, the MAC access time is the one that is difficult to calculate, since it depends on the network traffic and other network parameters. However, if the synchronization algorithm execute during the network startup, we may schedule the synchronization process without concurrent network tasks, since other tasks, such as routing protocols, are based on the synchronization process. In this case, as shown in previous work [12], the MAC access time is between $2\mu s$ and $10\mu s$.

B. Joint Solution using UAV

The operation of the proposed solution is divided into two phases. The first one refers to the UAV, that transverse the sensor field. The second phase is related to the position computation and clock synchronization. When a node receives a message from the UAV, it calculate its distance from the UAV using the RSSI technique and store the position and timestamp of the UAV. When a node has at least four messages, it is able to calculate its 3D position and synchronize its clock.

Figure 1 illustrates the UAV flight plan. A flight plan contain the airplane route which is previously designed by the network designer. During the flight, the UAV broadcasts its position and timestamp after each *broadcast interval*. While the end of the route is not reached, the algorithm retrieves the next point where the UAV should move and then, the UAV flights to the specific point with a certain speed. It is important to highlight that the periodic broadcast executes in parallel with the UAV displacement over the sensor field.

The algorithm executed in the sensor nodes to calculate their position and synchronize their clock is described below. When a node receives a message from the UAV, it calculates the distance to the UAV using the RSSI technique described above. The node retrieves the UAV position from the received message and store the position and distance to the related position in a *reference set*. The UAV timestamp is stored in a *stamp set*. If the number of received positions is greater than 4, the node is able to compute its position and synchronize its clock using the *reference set* and *timestamp sep*. To compute its position, the node uses the least squares method and to compute its local time, the node makes an average of all received timestamps. Also, for each received timestamp, the function adds a pre-defined One Hop Synchronization error (OHS) that is the error related to the MAC access time and propagation time.

IV. Simulation Results

A. Scenario description

The proposed 3D Localization and Synchronization integrated solution is compared to the literature solutions that solve each problem individually. However, to do a better comparison, we integrated one literature solution to solve the localization problem with one literature solution to solve the synchronization problem. We carefully studied the literature solutions that could be easily integrated to solve the 3D localization and synchronization problems, and we identified the Recursive Position Estimation Algorithm (RPE) and the Flooding Time Synchronization Protocol (FTSP) as the most appropriate ones.

The main goal of our performance evaluation is to evaluate the proposed integrated algorithm considering the following important metrics: (i) estimation position error and (ii) synchronization error (μs). For this, we vary two important network parameters, which are: (i) Number of network nodes and (ii) Network density. To carry out these evaluations, we used the flight plan illustrated in Figure 1. The simulations parameters are presented in Table I. The communication range of the sensor node and the UAV is 50m. This was done to have a fair comparison with the literature algorithm.

The number of beacon nodes in the RPE-FTSP integrated literature solution varies from 25 up to 200. The beacon nodes are equipped with a GPS receiver. It is important to note that in our integrated solution, only the UAV is equipped with a GPS receiver. To calculate the monitoring area (x, y), we used the number of nodes (n) and the communication range r_c of the sensor nodes. The third dimension (z) for each node is a random number between 0 and 10 m.

We used the SinalGo v.0.75.3 [15] simulator to evaluate the algorithms and each scenario was replicated 33 times with different seeds for random number generation. In all results, the curves represent the mean values, whereas the error bars represent the confidence interval of 95%.

B. Number of Nodes

In this section we evaluate the solutions for different number of network nodes. For this analysis, we fixed the network density in 30.

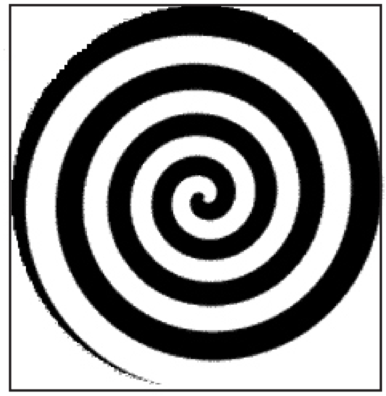


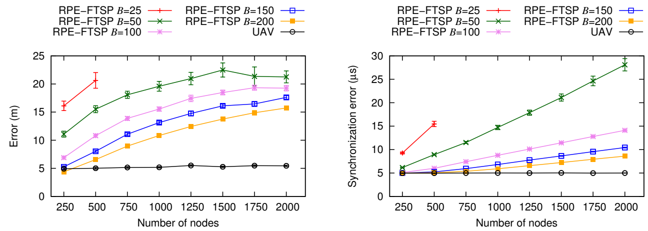
Figure 1. Flight plan.

Table I
Simulation parameters.

Parameters	Values
Number of nodes	250 to 2000
Density	15 to 50
Communication range	50m
UAV communication range	50m
UAV speed	10m/s
RSSI error	5%
OHS error	5μs
RPE-FTSP	25 to 200 beacon nodes
Monitoring area (x and y)	$x = y = \frac{n \times \pi \times r_c^2}{Density}$
Terrain (z)	0 to 10m
Flight altitude	20 to 50m
Broadcast interval (UAV)	1/second

Figure 2(a) shows the position estimation error. The proposed system has a small error in the position estimation and the error is not affected by the number of nodes, which is not observed in the RPE-FTSP algorithm. The RPE-FTSP position estimation error is around $3\times$ greater compared to our proposal and increases when we increase the number of nodes. This happens because by fixing the number of beacons and increasing the number of nodes, the unknown nodes estimate their position based on reference nodes, which has an estimated position. Thus, the estimation error spreads in the network. We also can observe that when we increase the number of beacon nodes, the estimation error decreases, since more unknown nodes will estimate their position using beacon positions. It is important to point out that, when we have only 25 beacon nodes, the RPE-FTSP algorithm is not able to estimate any position when $n > 500$. The main disadvantage of using many beacon nodes is the network cost, which increases substantially because of the GPS receivers. Also, when the localization and synchronization problems are solved, the beacon nodes become useless, since this process runs just once during the network lifetime.

The synchronization error is shown in Figure 2(b). When we increase the number of network nodes, the synchronization error of the RPE-FTSP algorithm also increases. This is due to the same fact that the RPE-FTSP position estimation error increases when we increase the number of nodes, since both



(a) Position error (b) Synchronization error

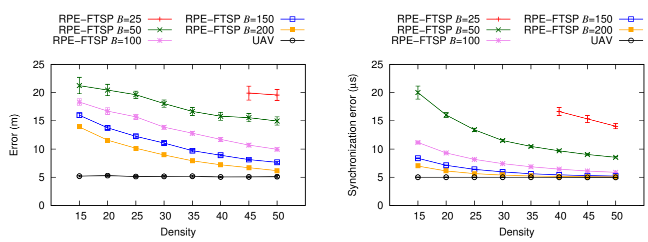
Figure 2. Number of nodes.

algorithms are executing together. When the network has 2000 nodes, the synchronization error of the RPE-FTSP algorithm is 1.89 times greater compared to the proposed solution (when $B = 200$). We also can see that the proposed synchronization system is not affected by the number of nodes.

C. Network density

This section evaluates the algorithms for different network densities. For this analysis, we fixed the number of network nodes in 750.

Figure 3(a) shows the error in the position estimation process. We can observe that the higher the values for the network density, the better is the RPE-FTSP performance. This is due to the fact that when we increase the network density for a fixed number of nodes, the monitoring area decreases. In this case, the position estimation error does not spread to many nodes. Our solution, that uses an UAV is not affected by the network density, since the UAV transverses all monitoring area.



(a) Position error (b) Synchronization error

Figure 3. Network density.

The same behavior is observed in the synchronization problem, since both algorithms runs together (Figure 3(b)). It is important to note that for high values of network density, there is no difference between our approach and the RPE-FSTP algorithm.

V. Conclusions

In this paper, we proposed a joint solution for the 3D localization and synchronization problems in WSNs by using an UAV. The UAV traverses the sensor field broadcasting its geographical position and clock time, allowing the sensor nodes to estimate their position and global time. Simulation results show that the proposed solution leads to a smaller synchronization and localization errors when compared to existing solutions. Moreover, the efficiency of our solution is independent of the number of nodes in the network, which is an important aspect in the case of scalability. Finally, under our solution, all the sensor nodes are able to calculate the global time and their position. As future work we intend to consider different flight plans, and conduct experiments in a real environment.

Acknowledgment

This work is partially supported by Canada Research Chairs Program, NSERC, DIVA Strategic Networks, CNPq, FAPESP (2012/22550-0 and 2013/05403-6) and FAPEMIG (APQ-01947-12).

References

[1] L. Villas, A. Boukerche, R. de Araujo, and A. A. F. Loureiro, "Highly dynamic routing protocol for data aggregation in sensor networks," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, 2010, pp. 496–502.

[2] L. A. Villas, A. Boukerche, H. S. Ramos, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, "Drina: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 676–689, 2013.

[3] L. A. Villas, A. Boukerche, D. L. Guidoni, H. A. de Oliveira, R. B. de Araujo, and A. A. Loureiro, "An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks," *Computer Communications*, pp. –, 2012.

[4] "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[5] L. Villas, A. Boukerche, D. L. Guidoni, H. A. Oliveira, R. B. Araujo, and A. A. Loureiro, "Time-space correlation for real-time, accurate, and energy-aware data reporting in wireless sensor networks," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '11. New York, NY, USA: ACM, 2011, pp. 59–66. [Online]. Available: http://doi.acm.org/10.1145/2068897.2068911

[6] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *The 9th International Conference on Network Protocols*, November 2001, pp. 35–41.

[7] H. A. B. F. Oliveira, A. Boukerche, E. F. Nakamura, and A. A. F. Loureiro, "Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm," *Perform. Eval.*, vol. 66, no. 3-5, pp. 209–222, 2009.

[8] D. Niculescu and B. Nath, "Ad hoc positioning systems (aps) using oao," in *IEEE INFOCOM*, November 2003, pp. 1734–1743.

[9] D. L. Guidoni, L. A. Villas, F. S. H. Souza, H. A. B. F. de Oliveira, and A. A. F. Loureiro, "A small world approach for scalable and resilient position estimation algorithms for wireless sensor networks," in *MOBIWAC*, 2012, pp. 71–78.

[10] K. Sinan Yldrm and A. Kantarc, "Time synchronization based on slow flooding in wireless sensor networks," *IEEE Trans. onParallel and Distributed Systems*, no. 99, 2013.

[11] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2011.

[12] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *SenSys '04*, 2004, pp. 39–49.

[13] A. Boukerche, *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press, 2008.

[14] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[15] Sinalgo, "Simulator for network algorithms," 2008, distributed Computing Group - ETH-Zurich.