Real-Time Relative Positioning with WSN

Rafael G. Aranha
Instituto de Telecomunicações,
Instituto Superior Técnico,
Technical University of Lisbon, Portugal
rafael.aranha@tagus.ist.utl.pt

Rui M. Rocha
Instituto de Telecomunicações,
Instituto Superior Técnico,
Technical University of Lisbon, Portugal
rui.rocha@lx.it.pt

Abstract

This paper addresses a scenario where a Wireless Sensor Network (WSN) is used to implement a real-time relative positioning system. Its goal is to obtain relative coordinates of objects (the WSN nodes) in which the nodes are generally moving relatively to each other. To accomplish this, the distance among nodes is measured and its relative position is computed allowing the representation of their spatial distribution.

The test platform uses Crossbow's motes to which special ultrasound (US) boards have been adapted. The US is thus the process used to measure distances in this system. Some of the issues addressed by this work are: precise time-synchronization between nodes, distance measurement procedure, positioning algorithms and its implementation on the TinyOS, Operating System used in the Crossbow's motes. Tests have shown that object positioning could be obtained with errors below 2% and using a computing time of about 1s per network node, on average.

1. Introduction

The fast evolution of wireless networks and the continuous development of microelectronics allowed the use of tiny microcomputer systems capable of processing data collected from different types of sensors in embedded network applications where information is transmitted from node to node until it reaches a central node where it is gathered, analyzed and eventually visualized. Such Ad-Hoc organization, optimized for embedded applications and comprising usually a high number of small, low-power, short-range sensor nodes, is know as a Wireless Sensor Network (WSN) [1].

The main goal of this work is to develop a system capable of operating in an indoor scenario where the localization of every node relatively to others is to be determined. Some nodes may have absolute geographical references and can thus be used as anchors so that relative coordinates may be converted to absolute ones. However, the main feature of the system is its ubiquitous nature, i.e. the fact that it doesn't restrict node's mobility allowing for real-time tracking of node's movements. The sprouting of new nodes or group of nodes will enhance the position estimation, as the new nodes become part of the network.

The paper is organized as follows. Section 2 makes a review of the main aspects related with positioning problems, highlighting the techniques, technologies and algorithms relevant for approaching the localization issue. Section 3 discusses the developed algorithms and solutions used in the development of our system, while Section 4 analyses the results obtained through experimental evaluation of those algorithms. Finally, Section 5 presents the main conclusion of the work.

2. Positioning techniques and technologies

Several positioning system can be used ubiquitously. However, most of the solutions suffer from several common limitations:

Limited scalability;

Dimensions not adequate to ubiquitous applications;

Too expensive to be used in large scale;

The majority needs a great deal of processing power and energy, which are rather limitative for sensor-based implementations.

In [2] several methods and algorithms suitable for measuring distances and obtaining node's localization using multihoping techniques are presented. Different localization approaches differ in their assumptions of: network's topology, nodes' capacity, propagation models (radio or acoustic), energy and timing requirements, environment scenarios (e.g. indoor or outdoor), node density, synchronization constraints, mobility, types of sensors and their corresponding sensing accuracy.

This work was funded by FCT – Foundation for Science and Technology.

978-0-7695-3330-8/08 $25.00 © 2008 IEEE
DOI 10.1109/SENSORCOMM.2008.54

2.1. Distance Measurement

Localization of objects often relies on determining the distance from these objects to different referential points with well known geographic coordinates. Therefore, distance measurement is of key importance in positioning systems.

Some of the basic measurement techniques for assess the distance between two objects are the following:

Time of Arrival (TOA): based on the measurement of the time a signal takes to travel from a source node to a destination node. The most well known example of this technique is GPS. Its major limitation is the requirement for an accurate synchronization between nodes.

Time Difference of Arrival (TDOA): based on the time a signal takes to arrive to a destination node relatively to a given time reference, e.g. the relative travel time of sound when compared to light. AHLos [3] and CRICKET [4] are two examples of systems that use this technique.

Angle of Arrival (AoA): allows for the determination of the angle from where the measurement signal was received by a node. Despite not being a direct distance measurement method it usually complements TOA and TDOA techniques even though it requires a high number of sensors to cover a 360º angle range. An example of such system is the APS [2].

Received Signal Strength Indicator (RSSI): based on the propagation characteristics of radio signals namely on the relation between the received signal power and the corresponding travel distance [5][6]. Its main limitation has to do with the non-linearity of the real propagation phenomenon, which depends on several factors such as: temperature, relative humidity, multipath fading, shadowing, etc. Examples of such systems are Microsoft's RADAR and UW SpotOn [7].

MIT's CRICKET and PARC's STAM [8] are example of projects that implemented positioning system although not seeking a solution for real-time relative localization of moving WSNs.

2.2. Positioning Algorithm

Most of positioning algorithms present results obtained through simulation based on ideal propagation models and on a high number of nodes that cannot be easily characterized experimentally. Their classification can be done according to the type of localization used:

Range-Based: use of sensors (e.g. sound) or radio signal to estimate distances;

Range-Free: position (location) is simply evaluated through the information about the nodes' connectivity. Some examples are [9]: Centroid, DV-Hop, Amorphous and Active Badge developed by USC/ISI, Rutgers and AT&T, respectively.

Positioning algorithms may also be classified in terms of computation types:

Centralized: all the required data has to be collected in a central node (e.g. sink node). Aside from the intrinsic disadvantages, a central node is able to produce an improved prediction of nodes' positions since it has access to a large amount of data;

Distributed: Communication and processing is restricted to a limited area independently of a critical central node;

Localized: each node seeks to obtain its own position.

A positioning algorithm able to accomplish the precision requirements of our system cannot be based on a *Range-Free* algorithm due to their usually high inaccuracy when used with small networks (less than 10 nodes). The choice is then restricted to the *Range-Based* type. Additionally, due to the reduced node's computation power, a centralized approach having a sink node in charge of the position evaluation of the remaining network is almost mandatory. However, as the sensor platforms continue to evolve, it will soon be possible to rely on a distributed approach to compute the position of a sub-group of nodes.

Most of the analyzed algorithms use geometrical calculations based on successive triangulations or multilaterations to assess a position relatively to some reference points. These methods may propagate possible measurement errors even though they use minimization techniques so typical in these algorithms.

Under these circumstances, we selected the *MultiDimensional Scaling (MDS)* [10][11] as the one who would best fit our system's requirements. MDS is a set of data analysis methods having the goal of estimating the relation among certain objects. The raw data, analyzed by MDS, are typically likelihood or unlikelihood measurements between objects. The analysis result is a spatial configuration of the objects represented by points. Similar objects, i.e. having a great deal of likeliness, are represented quite close to each other.

2.3. Time Synchronization

In this project, the need for time synchronization comes from the fact that it is required to measure the time that a US burst takes to go from a source node to destination nodes. However, random delays during the sending and receiving procedures may produce errors that impact significantly the measurement precision and need to be analyzed and mitigated [12]. These effects are due to software or hardware delays associated with Application, MAC and PHY layer's functionality. These time delays may be divided into:

Sending Time: time spent in message formatting and transmission to the MAC layer of the sender. It depends on the OS primitives as well as on the current processor load;

Access Time: elapsed time from the moment of the first trial to access the media until the start of message transmission on the air. It's dependent on the MAC and on the network load, as well;

Transmission Time: time required for the PHY layer to completely deliver the message onto the air. It depends on the message size and on the bit rate;

Propagation Time: time that a message takes to go from sender to receiver.

Receiving Time: elapsed time from message processing until it is handed over to the application on the receiver side; same characteristics as the sending time.

As referred in [13], the WSN's node programming facilities provide direct access to the MAC layer allowing the time stamping or the time readout of messages (referred to the current value of local clock) in the transmission and reception, respectively. In this manner it is possible to ignore the sending, access and receiving times. Disregarding the propagation time, as the nodes are always quite close to each other, only the transmission time actually needs to be determined.

3. System Implementation

The system comprises a given number of moving nodes plus a base station, also mobile (likely a PDA). The base station mission is to collect data from the network and process it.

After receiving the collected data these will be processed using algorithms for calculating and representing the relative positioning of the network nodes.

3.1. Platform

We have used Crossbow motes to which we have attached a special US daughter board for measuring the distance between two nodes in a real world environment. This daughter board is a SRF08 US module, from Devantech. This module has an US emitter and an US receiver, which can compute distances to obstacles 5m away. This module communicates with the mote using an I2C interface. Its microcontroller (PIC 16F872) code had to be modified in order to allow sending/receiving US impulses between different nodes and not only within the same board. In Figure 1 it is shown a WSN node´s prototype with an adaptation board designed to interface the US module with the Crossbow's mote. A smaller cone is standing on the top of the emitter and a bigger one on the top of the receiver for sending and receiving US pulses respectively, increasing the coverage angle of the US devices.

Figure 1. Mote with an Ultra-Sound daughter board.

3.2. Ultra-Sounds

The precision of the technology used to measure the distance between devices is the critical factor of a positioning system. After analyzing the measurement techniques seen before the one that presents itself as the simpler for a WSN is the TDOA.

The use of acoustic signals was, as presented in [14][15] and as experimentally tested, considered less accurate. The distance between two motes using the sound sensor and microphone from its own sensor board [16] in a way that they can ear each other is limited to around 1m. To filter the signal, sample it and compute its average is possible in a static system or network but not in a system where nodes are moving around and where one sample should be enough to get the desired distance.

Therefore, the use of US reduces the influence of the external factors seen before. One disadvantage is that US are very directional which prevents them from being used within a 360º coverage range. Systems like AHLos use eight US sensors which allow them not only to compute the distance but also to use the AOA technique. On the other hand it makes the system more complex and larger. The choice in this project has been to use an off-the-shelf US module, adapt it to our needs and to design a board to interconnect it to the mote. Later on, a dedicated daughter board, based on this experience, was developed and put to use in another project [17].

The limitation regarding the coverage angle of the US was circumvented using reflection cones as referred before.

3.3. Time Synchronization between motes

Recalling the initial goal, it was intended that both the sender and receiver motes' US sensor shot and listened at exactly the same time, so that the sound's measured propagation time could be directly related with the distances between the motes.

To accomplish the proposed scheme the following algorithm was implemented:

The emitter node sends a radio message telling that it will send an US pulse within *k* milliseconds;

After the message being sent an event is triggered in the emitter node which will request to the MAC component the clock value of when the message went out - $Tx_{MACClockValue}$

In the emitter node a timer is started at $Tx_{ClockValue}$ which will timeout after $Alarm_{Tx}$ seconds and trigger the emission of an US pulse, being:

$$Alarm_{Tx} = k - (Tx_{ClockValue} - Tx_{MACClockValue}) \quad (1)$$

When the receiver nodes start to receive the radio message sent by the emitter node, at $Rx_{MACClockValue}$, they initiate an alarm which will trigger $Alarm_{Rx}$ seconds after:

$$Alarm_{Rx} = k - (Rx_{ClockValue} - Rx_{MACClockValue}) \quad (2)$$

$Rx_{ClockValue}$ is the time when the $Alarm_{Rx}$ is triggered. In order to compensate for the Receiving Time delay the difference from the two clock values is subtracted from *K*. Hence there is no need for synchronization between clocks.

Considering the time it takes for a radio signal to propagate for a few meters can be ignored:

$$Alarm_{Rx} = Alarm_{Tx} \quad (3)$$

This means that the alarms in the emitter node and at the receiver nodes will rise at the same time and trigger the emitter and the receiver US sensors all at once thus eliminating other interferences and delays.

3.4. Localization

The localization solution that we aim to accomplish is: after being able to get at least one approximate measure between all the points or even between just a few, it is intended to graphically represent the relative position between all the points of the network.

The following steps describe the implementation of the used algorithm - Multidimensional Scaling (MDS).

A proximity matrix **D** is filled with the known measured distances between the **N** nodes in a network.

The double-centering method is applied, in which the matrix is centered around the average value of its entries, thus obtaining the **B** matrix:

$$B = -\frac{1}{2}\left(I - \frac{1}{N}U\right)D^2\left(I - \frac{1}{N}U\right) \quad (4)$$

being **I** an identity matrix NxN and **U** an NxN matrix of 1s. Next the eigenvalues (matrix **A**) and the eigenvectors (matrix **V**) of the B matrix are computed using (5).

$$B = VAV^T \quad (5)$$

The **m** largest positive eigenvalues and its correspondent eigenvectors are retained, being **m** the intended number of spatial dimensions (2D, m=2).

The spatial distribution which corresponds to the node's coordinates relatively to each other is obtained from the computation of the **X** matrix:

$$X = VA^{\frac{1}{2}} \quad (6)$$

To obtain an absolute positioning it would be necessary to know the absolute position of some of the nodes. Therefore the map that is shown could be rotated relatively to its real position.

From what was said one can infer that there is a need to know all the measures between all the nodes of a network in order to fill in the proximity matrix. When some of the distances between some of the nodes aren't measured, those entries are considered to be zero in the proximity matrix **D.**

An iterative process is then used in order to minimize the differences between the computed measures and the real ones:

$$\sigma = \sum_{i<j} W_{i,j}\left(d_{ij}(X) - \delta_{ij}\right)^2 \; 2 \quad (7)$$

being d_{ij} the Euclidean distance between the node ***i*** and the node ***j*** taken from the **X** matrix, and δ_{ij} the distance initially measured and filled in the **D** matrix. If it is impossible to assess that distance, then $W_{i,j} = 0$.

$$\sigma(X[K+1]) - \sigma(X[K]) < \varepsilon \quad (8)$$

Thus it is intended to iterate the algorithm **K** times until ε has the possible smallest value.

It is therefore intended to obtain a final matrix **X** with all the nodes' coordinates computed from the distances that were possible to measure between the maximum number of nodes belonging to the network.

4. Experimental results

Tests were made to evaluate the hardware and software platforms already referred.

Initially measures were taken to determine the calculation errors of the synchronization algorithm and the delay of the US triggering. Two motes, A and B, were placed apart from each other with the following distances: 100, 300, 400 and 500cm.

Twenty measurements were done between mote A and B and vice-versa, in each of the previous mention distances. This measures where taken consecutively and alternately between the motes, with a 5 sec interval between each US shoot order.

The motes were at the same height and there were no obstacles between them. The scenario was a 7mx7m room with tables, chairs, computers and glass walls.

The orders that received no results, 10% on average, were excluded.

The experimental results are presented in Table I.

Table I

Real distances between mote A and B (100, 300, 400 e 500 cm) versus the distances obtained in result of the use of US boards.

	REAL DISTANCES VS MEASURED (cm)							
	100		300		400		500	
nº measurement	A to B	B to A	A to B	B to A	A to B	B to A	A to B	B to A
1	102	90	298	300	392	396	498	498
2	102	94	298	294	398	400	498	500
3	104	94	296	298	400	396	506	502
4	92	92	298	300	392	396	494	496
5	100	90	276	294	386	398	498	496
6	104	96	298	294	392	394	498	496
7	102	90	308	298	396	400	496	504
8	98	94	294	296	400	406	490	496
9	102	92	296	300	398	402	500	496
10	98	94	300	298	398	404	500	500
Average:	100,4	92,6	296,2	297,2	395,2	399,2	497,8	498,4
Error:	0,40%	7,40%	1,27%	0,93%	1,20%	0,20%	0,44%	0,32%
Standard Dev:	3,44093	2,009975	7,613147	2,4	4,308132	3,709447	3,944617	2,8

Table I shows, for each distance, the average and standard deviation values of the readings as well as the error to the real distance. For example, from B to A the highest error was achieved during the 1st, 5th and 7th reading, being 10cm in 100cm, i.e. 10%. Of course the measurement's quality requirement depends on the intended applications. Nevertheless, to achieve localization of objects or people inside a room these values are considered acceptable.

In a second stage the implementation of the localization algorithm was tested.

A small test-bed network with just three nodes, forming an equilateral triangle with a 200cm side, was setup.

The test was initiated by sending, from one node to the others, an order to start the process, i.e one node sends a radio message followed by an US pulse and then all the others reply to the sender with the measures they have computed. After all nodes have done the same cycle, the distances between the motes (in both directions) were obtained. This data was loaded into a matrix - Table II - to which the localization algorithm will be applied.

Table II

Distance matrix filled with the distances measured between the nodes of an equilateral triangle, in centimetres.

motes	**1**	**2**	**3**
1	0	192	182
2	192	0	192
3	182	192	0

Finally, the relative position of the nodes was computed and represented, as shown in Figure 2. The results obtained from the localization algorithm are an optimum solution of all the distances between the motes based on the measured distances given by the motes with the US boards.

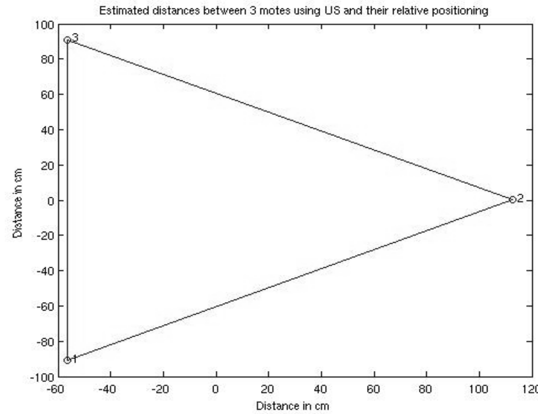

Figure 2. Estimated distances between 3 motes using US sensors.

As can be seen in Table II, the localization algorithm needs to use a symmetric matrix. Thus, either only one of the readings is taken into consideration (e.g. A to B and not B to A) or an average is computed using the two values. In the present scenario, the former has been chosen after taking into consideration the error values seen in Table I as well as the fact that if the motes are on the move both reading won't be necessarily the same.

In a third stage, the goal was to measure the time that took since the emission of the first order to the first mote to initiate the process until the computation and representation of the motes' positioning on the network.

Considering a network of N motes, the time period, Δt , that elapses since the emission of the order to the *mote* **i** to start a cycle of measurements until all the messages from the N-1 motes are received that tell the distance measured to the mote **i** is:

$$\Delta t = t_a + t_b + t_c + t_d \quad (9)$$

being,

t_a, the time it takes since an order is sent to mote **i** until this mote initiates the warning process to the other **N-1** motes that an US pulse will be sent;

t_b, the time that mote **i** waits until it sends the US pulse. In the implemented algorithm the motes are programmed to wait 1s;

t_c, the time the US pulse takes to trigger and propagate. The trigger time has an average of 600us and the propagation one, for a distance of 5 meters and c_{sound} = 344m/s of 14.53ms;

t_d, the time it takes since the **N-1** motes start to send the messages with the measured results until the distance matrix is filled;

As referred before, several sources of delay were analyzed in the transmission of the radio messages. Summing up, the uncertainty of the time period of the main sources is related to:

Sending and Receiving: from 0 to 100ms;

Medium access: from 10 to 500ms;

Transmission: 10 to 20ms.

Hence, t_a will have a maximum value around 620ms. t_d, being dependent of the number of nodes in the network due to possible collisions when trying to access de media, can reach some tens of a second. Experimentally it was observed that in the case of using 3 motes the average value of Δt is 3.19s. If t_b would be reduced, Δt could be reduced to 2.90s.

For a network of N motes, the time period since the beginning of the process until the motes' position representation is:

$$\Delta t_{total} = (N-1)\Delta t + t_{alg} \tag{10}$$

being t_{alg}, the time the localization algorithm takes to compute the relative localization of the nodes from the distance matrix. This value depends on the number of nodes in the network and on the measurement's error, which will originate a higher number of iterations in the localization algorithm in order to reach a possible solution.

It should be noted that, theoretically, to achieve the relative positioning between all the nodes of a network it is not necessary to measure all the distances between the motes. Ideally it would be enough if all the points in the graph would be connected at least by only one connection, which means that there would be no need for the distance matrix to be all filled.

In practice and due to measurement errors, a large number of measurements will achieve a more accurate representation of the nodes' relative positioning.

5. Conclusions

In this paper we have presented a real architecture, implemented with motes running TinyOS and using an US-based distance measurement technique, whose main goal was to attain the relative localization between several motes in a network, being this virtualized by persons or objects which are part of a sensor network.

With distances between two nodes going from 100cm to 500cm measurement errors around 1.52% were achieved, on average.

Using the Multidimensional Scaling algorithm and starting from a matrix filled with the measured distances gotten from the US motes it was possible to compute and represent the relative positioning between those motes.

It was observed that the time it takes to complete a cycle of sending an US pulse from one mote and to receive the messages with the measured results from the other motes is around 3s.

For this architecture to be implemented in a real-time environment the two main limitations are: the time it takes for a sensor to trigger an US shot, and the time spent by each mote trying to access the media, which increases with the number of network nodes.

The MAC delay uncertainty can be minimized by other types of media accesses, namely of schedule-based nature. On the other hand, the US limitations are more difficult to overcome. Some work to alleviate US shots' uncertainty, range, multipath effect, interferences and coverage lies still ahead. However, with the emerging IEEE 802.15.4a standard its Ultra-wideband [18] physical layer and its applications to the accurate measurement of distances, one can foresee an easier implementation of such positioning systems.

References

[1] A. Bharathidasan, V. Ponduru, *Sensor Networks: An Overview*, Department of Computer Science - University of California, 2004.

[2] Dragos Niculescu, *Positioning in Ad Hoc Sensor Networks*, IEEE Network, July/August 2004, pp. 24-29.

[3] A. Savvides, C. C. Han, M. B. Srivastava, *Dynamic FineGrained Localization in AdHoc Networks of Sensors*, Proceedings of MOBICOM '01, 2001, July 2001.

[4] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan,, *The Cricket Location-Support System*, 6th ACM MOBICOM, Boston, MA, August 2000.

[5] P. Bergamo, G. Mazzini, *Localization in Sensor Networks With Fading and Mobility*, DI, University of Ferrara, 2001.

[6] N. Patwari, A. Hero III, *Location Estimation Accuracy in Wireless Sensor Networks*, University Michigan, 2002.

[7] Jeffrey Hightower, Gaetano Borriello, *Location Systems for Ubiquitous Computing*, Computer IEEE, August 2001.

[8] Y. Zhang, L. Ackerson, D. Duff, C. Eldershaw, *STAM: A System of Tracking and Mapping in Real Environments*, IEEE Wireless Communications, December 2004.

[9] Tian He, Chengdu Huang, Brian M. Blum, *Range-Free Localization Schemes for Large-Scale Sensor Networks*, MOBICOM 2003.

[10] K. W. Cheung, H. C. So, *A Multidimensional Scaling Framework for Mobile Location Using TOA*, IEEE Transactions on Signal Processing, Volume: 53 , Issue: 2, February 2005.

[11] X. Ji, H. Zha, *Sensor Positioning in Wireless Adhoc Sensor Networks using MDS*, INFOCOM '04, 2004.

[12] Kay Römer, Philipp Blüm, Lennert Meier, *Time Synchronization and Calibration in Wireless Sensor Networks*, Wiley, 2005.

[13] S. Ganeriwal, R. Kumar, M. Srivastava, *TimingSync Protocol for Sensor Networks*, ACM SenSys Conference, November 2003.

[14] L. Girod, V. Bychkovskiy, J. Elson, D. Estrin, *Locating tiny sensors in time and space: A case study*, University of California, 2002.

[15] C. Whitehouse, The Design of Calamari: an Adhoc Localization System for Sensor Networks, University of California, Fall 2002.

[16] Mica Sensor Board, Available: h*ttp://today.cs.berkeley.edu/tos/hardware/hardware.html*, XBow, 2005.

[17] L. Pedrosa, P. Melo, R. Rocha, R. Neves, *A Flexible Approach to WSN Deployment*, accepted in The First International Workshop on Sensor Networks (SN 2008), August 2008.

[18] R. Szewczyk, UWB: Technology and implications for sensor networks, NEST Meeting, August 2004.