Engineering a Search and Rescue Application with a Wireless Sensor Network - based Localization Mechanism

Ricardo SEVERINO, Mário ALVES
IPP-HURRAY! Research Group, Polytechnic Institute of Porto (ISEP/IPP)
431, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal
rars@isep.ipp.pt, mjf@isep.ipp.pt

Abstract

The advent of Wireless Sensor Network (WSN) technologies is paving the way for a panoply of new ubiquitous computing applications, some of them with critical requirements. In the ART-WiSe framework, we are designing a two-tiered communication architecture for supporting real-time and reliable communications in WSNs. Within this context, we have been developing a test-bed application, for testing, validating and demonstrating our theoretical findings - a search&rescue/pursuit-evasion application. Basically, a WSN deployment is used to detect, localize and track a target robot and a station controls a rescuer/pursuer robot until it gets close enough to the target robot. This paper describes how this application was engineered, particularly focusing on the implementation of the localization mechanism.

1. Introduction

Recent advancements in wireless communications and micro-sensing embedded technologies are enabling the real deployment of Wireless Sensor Networks (WSN). WSNs offer new ways to monitor/control our environment in a continuous and almost invisible way, holding the promise of many new ubiquitous computing applications. Examples include wildlife habitat monitoring [1], precision agriculture [2] or intrusion detection and tracking [3].

At least a subset of these emerging and future applications is expected to impose real-time requirements. Thus, we are devoting our efforts to the development a two-tiered communication architecture for improving the timing and reliability behaviour in WSNs, in the ART-WiSe framework [4,5].

In order to test, validate and demonstrate our theoretical findings, we have triggered the development of a test-bed application [6]. We have opted for a search&rescue/pursuit-evasion application scenario, since it imposes stringent timing requirements to the underlying communication infrastructure. Basically, a control station detects, localizes and tracks a target robot (robot in distress/intruder) and controls a rescuer/pursuer robot until it gets close enough to the target robot. The WSN motes are used for the localization (RSSI-based) and for communication between the different entities involved.

This paper focuses on how this test-bed application was engineered. Namely, we explain why and which localization mechanism was chosen as well as how it was implemented. An extended version of this paper is presented in [7].

The paper is structured as follows. Section 2 presents an overview of the test-bed application and of the used technologies. Section 3 addresses the choice of the localization mechanism and how it was implemented. Finally, section 4 concludes this paper and provides some directions for future work.

2. Test-bed application overview

2.1 Snapshot of the test-bed application

The overall objective of the application is to detect, localize and rescue a target entity, within a certain region covered by a WSN deployment. Mobile robots are currently being used to act as target and rescuer/pursuer entities.

The target robot is supposed to be in distress (search&rescue context) or to be an intruder (pursuit-evasion context). A rescuer/pursuer robot will be instructed by a control station to navigate towards the target robot, until it gets close enough to it. Figure 1 illustrates an example scenario.

The target robot movement is remotely controlled by an operator, using a joystick. A WSN node mounted on top sends periodic messages to signal its presence, which are then relayed by the WSN to the Control Station. The Control Station implements a 3D virtual display of rescuer and target robots status, based on messages sent from the rescuer robot and on the target localization mechanism, respectively. It also displays relevant information related to the rescuer robot status:

position, heading, mission status, waypoints. The Control Station then computes the target robot location and informs the rescuer robot that will immediately initiate its mission by moving towards the last known position of the target robot in an autonomous fashion. This process is repeated until the rescuer robot is close enough to the target robot.

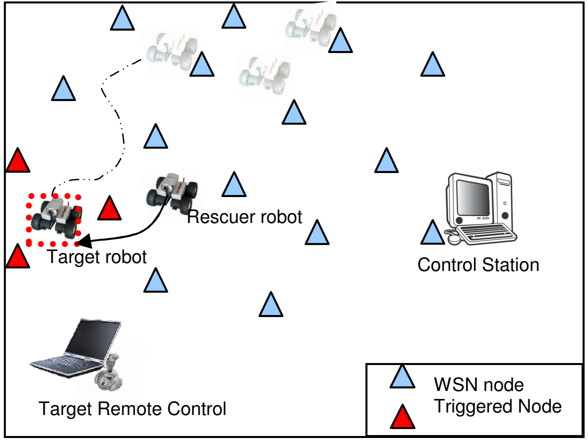

Figure 1. Snapshot of the application

Figure 2 presents the current test-bed deployment, showing a rescue mission in progress.

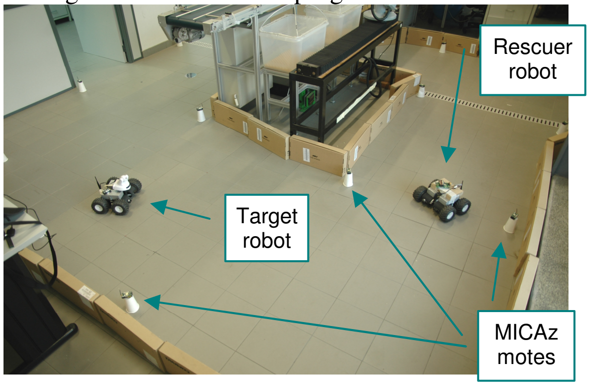

Figure 2. Picture of the current indoor deployment

While currently only one target robot and one rescuer/pursuer robot are being used, the number sensor nodes and the number of robots in each "team" will tend to grow in the medium-term.

2.2 Outline of the used technologies

The WSN nodes are MICAz motes [7], featuring an ATMEL ATmega128L 8-bit microcontroller with 128 KB of in-system programmable memory. In order to interface the rescuer robot and the WSN, a MIB510 interface board [9] (with a MICAz plugged in) providing RS-232 communication was used (Figure 3).

The MICAz nodes run TinyOS [10], an open-source event-driven operating system designed for WSN nodes with limited resources. These nodes were chosen since they feature an IEEE 802.15.4-compliant radio transceiver, which is the federating communication protocol for the lower tier of the ART-WiSe architecture. In this context, we have developed a set of tools, including an open source implementation of the ZigBee/IEEE 802.15.4 protocol stack [11].

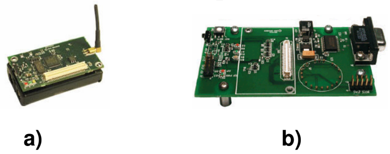

Figure 3. MICAz mote (a) and MIB510 interface board (b)

The mobile robotic platform used in the application is the WifiBot [12]. The system architecture is build around a double Ethernet-I²C bus and a CPU – 4G Access Cube [13], a hardware platform dedicated to Wireless LAN Mesh Routing –acting as an interface between the two. The 4G Access Cube also features an IEEE 802.11 access point.

3. On the localization mechanism

3.1 Choosing the localization mechanism

In this kind of applications, the availability of a localization mechanism is crucial for tracking of the target and rescuer robots. The need for absolute positioning system is also evident, since it is not feasible to merely rely on relative positioning methods (i.e. odometry) when considering long distances, due to the cumulative positioning error problem.

Assuming that a WSN has already been deployed in a certain region, it is interesting to use the WSN capabilities to provide localization functionality, in addition to other services such as monitoring/control. Moreover, with such a WSN-based localization mechanism, the application is not dependent on proprietary localization technologies (e.g. GPS) which could be too costly in a large scale WSN.

There are many proposals on localization techniques based on radio range measurements, like Time of Arrival (ToA) or Radio Signal Strength (RSS) like the Cricket [14] and MoteTrack [15]. However, many of these methods either involve special hardware development or are very time consuming. More detail on these methods can be found in [17].

We have opted for the RSS range measurement method since it does not involve any additional/special hardware design and it can be easily implemented in the MICAz motes by using the CC2420 [18] Radio Signal Strength Indicator (RSSI) function.

Since we are estimating distances, the method to compute position might be Lateration. However, this method imposes some practical problems like its high sensitiveness to distance measurements errors, and the fact of being computationally expensive. A much simpler method – Min-Max – was proposed by Savvides et al.[19] as part of the N-hop multilateration approach. The main idea is to construct a bounding box for each anchor using its position and distance estimate, and then to determine the intersection of these boxes. The position of the node is set to the centre of the intersection box (Figure 4).

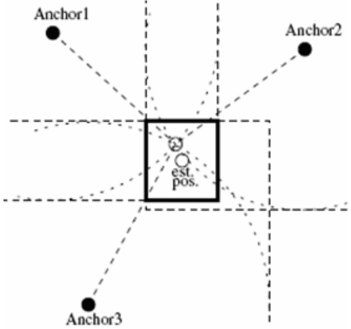

Figure 4. Estimated position (Min-Max and Lateration) [19]

Figure 4 illustrates the Min–Max method for a node with distance estimates to three anchors. Note that the estimated position by Min–Max is close to the (real) position computed through Lateration (i.e., the intersection of the three circles), which is also represented in the figure.

The bounding box of an anchor *a* is created by adding and subtracting the estimated distance (*da*) from the anchor position (*xa*, *ya*) (1):

$$[x_a - d_a, y_a - d_a] \times [x_a + d_a, y_a + d_a] \qquad (1)$$

The intersection of the bounding boxes is computed by taking the maximum of all coordinate minimums and the minimum of all maximums (2):

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \qquad (2)$$

The final position is set to the average of both corner coordinates.

3.2 Implementing the mechanism

In order for the carry out the implementation of the localization mechanism, a set of experiments had to be performed. The objective was to build an algorithm that could relate distance (to the anchor node) to the obtained RSSI measurements, for later processing using the Min-Max algorithm in the Control Station and in the rescuer robot. The most important aspects of this experimental work are presented in [21]. In summary, we have evaluated the most adequate MICAz transmission power level to our environment and found out which intervals of RSSI values were expected at different distances, thus enabling RSSI to distance conversion. This was done by establishing a correspondence between discrete range levels and the spread of RSSI values encountered for that same range. Figure 5 shows the plot of Distance vs. RSSI for power levels 3, 4 and 5 of the MICAz, ranging from -25 dBm to -15 dBm.

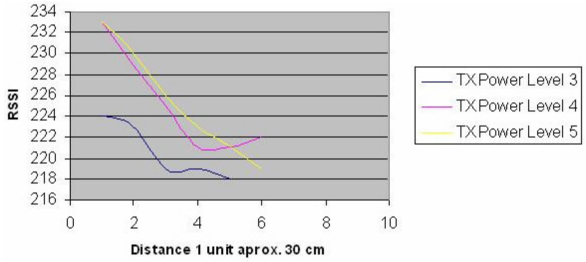

Figure 5. Distance vs RSSI plot

Two different approaches were implemented, based on the same localization method – one for the rescuer robot and the other one of the target robot. Figure 6 provides some intuition on these implementation approaches.

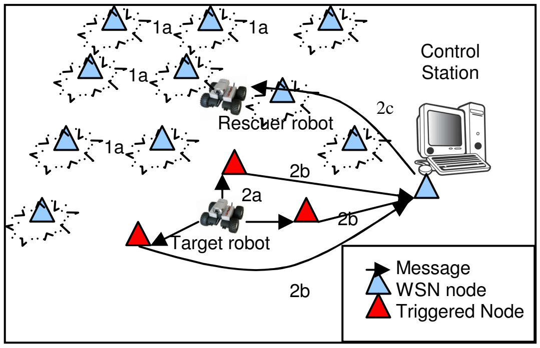

Figure 6. Messages for the localization mechanism

For the rescuer robot positioning mechanism, every WSN node periodically broadcasts a message which contains the node (*x*,*y*) coordinates and address. Some of these messages are received by the mote on the rescuer robot that processes the information and computes its position. In both implementations, only the three nodes with the strongest RSSI are used in the computation, in order to reduce uncertainty.

The target robot detection mechanism and the subsequent mission dispatching to the rescuer robot are illustrated in the timing diagram of Figure 7.

The target robot initiates the process by announcing its presence by sending a distress ("help") broadcast message (2a) at a pre-programmed transmission power and periodicity. Every WSN node that receives that message stores the received RSSI and sends a "Distress Alert message" containing that value and its coordinates to the control station (2b). The Control Station is expected to receive multiple "Distress Alert messages" from different nodes. As soon as a sufficient number of messages is received (e.g. 7 messages) the target robot's position is computed based on the same algorithm used for the rescuer robot positioning.

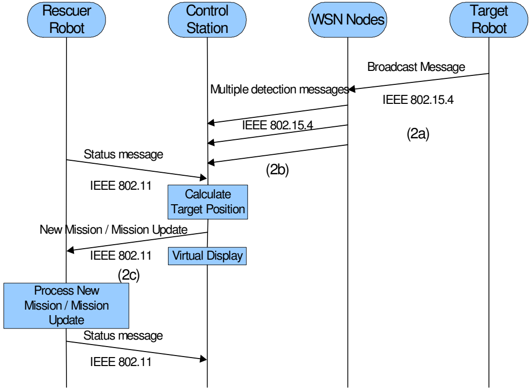

Figure 7. Timing diagram of the messages involved in the target detection and localization mechanisms

The Control Station then instructs the rescuer where to go (2c) (e.g. last known position of the target robot), via IEEE 802.11. This process is repeated iteratively until the rescuer robot reaches the target robot.

The localization mechanism presented a maximum error of approximately 70 cm. We did not expect better results for the localization mechanism since, as stated in [22], there are many sources of RSSI variability that may cause errors in the RSSI measurements, eventually leading to the computation of a wrong position.

4 Conclusions and future work

This paper outlines the most important aspects related to the engineering of a search&rescue application, where the mobile entities (robots) are located using the underlying communication infrastructure – a Wireless Sensor Network (WSN). We particularly focus on the theoretical and practical aspects related to the implementation of a localization mechanism based on radio signal strength measurements. This practical experience on the technology under use in our test-bed is of paramount importance for future developments.

We have shown that it is possible to exclusively rely on the WSN for providing a simple yet effective localization scheme. Nevertheless, some improvements regarding the application design were identified, like the computation of the target robot localization in the WSN (for a faster identification of the target's position) and the use of odometry information for improving the accuracy of the localization mechanism.

Future work concerning the ART-WiSe test-bed application will address issues such as the extension of the WSN deployment, enlarging the rescuer and target robot teams and the integration of our IEEE 802.15.4/ZigBee protocol stack in the MICAz motes, for supporting multi-hop real-time communications in the WSN.

5 References

[1] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. "An analysis of a large scale habitat monitoring application". In Proceedings of the Second ACM conference on Embedded Networked Sensor Systems (SenSys), 2004.

[2] J. Burrell, T. Brooke, and R. Beckwith. "Vineyard computing: Sensor networks in agricultural production". IEEE Pervasive computing, 3:38–45, 2003.

[3] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y-R. Choi, A line in the sand: "A wireless sensor network for target detection, classification, and tracking". Computer Networks (Elsevier), 46(5):605–634, 2004.

[4] A. Koubaa, M. Alves, "A Two-Tiered Architecture for Real-Time Communications in Large-Scale Wireless Sensor Networks: Research Challenges", WIP Proceeding of the 17th Euromicro Conference on Real-Time Systems (ECRTS'05), Palma de Mallorca, Spain, July 2005.

[5] http://www.hurray.isep.ipp.pt/art-wise

[6] M. Alves, A. Koubaa, A. Cunha, R. Severino, E. Lomba, "On the Development of a Test-Bed Application for the ART-WiSe Architecture", WiP Session of the 18th Euromicro Conference on Real-Time Systems (ECRTS'06), Germany, July, 2006.

[7] R. Severino, M. Alves, "Engineering a Search and Rescue Application with a WSN-based Localization Mechanism", IPP-HURRAY Technical Report TR-061201, December 2006.

[8] http://xbow.com/Products/productsdetails.aspx?sid=101

[9] http://xbow.com/Products/productsdetails.aspx?sid=79

[10] http://www.tinyos.net

[11] http://www.open-zb.net

[12] http://www.robosoft.fr/wifibot.html

[13] http://www.meshcube.org/index_e.html

[14] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System", ACM Sigmobile (Mobicom), 2000.

[15] K. Lorincz, M. Welsh, "MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking", Int. Workshop on Location and Context-Awareness (LoCA 2005) at Pervasive 2005, May 2005.

[16] A. Savvides, C.-C. Han, M. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors", 7th Int. Conf. on Mobile computing and networking, Rome, Italy, pp. 166-179, 2001.

[17] J. Gibson, "The Mobile Communications Handbook", IEEE Press 1999.

[18] Chipcon, SmartRF CC2420 Datasheet (rev 1.3), 2005.

[19] A. Savvides, H. Park, M.Srivastava, "The bits and flops of the N-hop multilateration primitive for node localization problems", in First ACM Int. Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA, 2002.

[20] Langendoen, N. Reijers, "Distributed localization in wireless sensor networks: a quantitive comparison" in International Journal of Computer and Telecommunications Networking, Special issue: Wireless sensor networks, November 2003.

[21] R. Severino, M. Alves, "On a Test-bed Application for the ART-WiSe Framework", IPP-HURRAY Technical Report TR-061103, November 2006.

[22] D. Lymberopoulos, Q. Lindsey, A. Savvides, "An Empirical Analysis of Radio Signal Strength Variability in IEEE 802.15.4 Networks using Monopole Antenas", Yale University ENALAB TR 050501.