BAT
Ultrasonic positioning system

Jorge Francisco Catarino Heleno
Department of Computer Science and Engineering
Instituto Superior Técnico
Advisor: Prof. Alberto Manuel Ramos da Cunha

Keywords: Indoor positioning; Wireless sensor networks; Distributed systems; Arduino

Abstract: Sensor networks are the key to gathering the information needed by smart environments, whether in buildings, industrial, home, shipboard, transportation systems automation, or elsewhere. One of the most important type of information is positioning. This type of information is normally very precious since it is hard to gather and people are not willing to give it away easily. However, in these types of systems, which gather information, it is usually impractical running wires or cabling. A sensor network is required to be fast as well as easy to install and maintain. With the enormous technological growth in the wireless technology field, it is possible to make these kind of systems function in a completely cable free environment, therefore being able to operate in virtually any kind of environment. In this document we present BAT, an ultra-sonic indoor positioning system, an Arduino powered wireless sensor network operating via Wi-Fi. This system is composed of three key components. The wireless sensor network itself, an aplication programming interface which the end-user will use to get information from the system, and a database storing the data of all the system.

1 INTRODUCTION

Outdoor Location Based System (LBS) has proven to be very successful while indoor LBS haven't fully developed due to the lack of an accurate positioning technology (Filonenko et al., 2013). Eventhough these systems already exist, most of them rely on external systems (such as the user's phone) to calculate such a position rendering the system dependent on external factors (other than the obvious physical ones).

In this work we present an independent, affordable system with a Wireless Sensor Network (WSN) using ultrasound technology and Time difference of arrival (TDOA) algorithms to calculate the position of an object. Positioning data is a very precious type of information which has been pursued by many. This happens because such an information is hard to gather (since normally people are not willing to provide it). In the past decades, wireless localization technologies have undergone considerable progress (Amundson and Koutsoukos, 2009), being now able to be performant in the localization process while indoors, fixing the flaws outdoor systems such as Global Positioning System (GPS) indoors. Also current advancements in the indoor location systems makes it possible to have a performant Indoor positioning system (IPS) nowadays as well as the proliferation of wireless localization whcih technologies provides a promising future for serving human beings in indoor scenarios (Zhang et al., 2010). Indoor location systems provide a solution to the inefficiency of current and widely used positioning systems such as GPS (Sugano et al., 2006). Given that Indoor location systems provide very important informations for businesses such as shopping centers, public agencies, etc.When designing a indoor location system we come up with some challenges due to the complexity of the environment itself such as None line of sight (NLOS), multipath effect and noise interference. Although this type of negative effects cannot be eliminated completely technologies advancements in the last years are constantly going to improve indoor human and object tracking. Surveys about these technologies can be found in (Amundson and Koutsoukos, 2009), (Zarocostas, 2008), (Buznă and Cernea, 1991). Even if we ignore the challenges already provided by today's indoor location technologies, we end up with some more challenges while building an IPS, such as the type of system being used. The type of calculations made in any of the existing technologies make it hard for a microprocessor with low resources (such as memory and CPU) to be performant (Yick et al., 2008), so in the end, to build this system we have to think and the de-

sign the architecture of such.

Recent advances in Microelectromechanical systems (MEMS) technology and wireless communications, it is possible to have low-cost, and multifunctional sensor nodes capable of handling wireless communications (Akyildiz et al., 2002) as well as retrieve lots of environment data such as light, noise, etc.

With such advances in this area it is possible to solve some problems of the systems used nowadays using known techniques and applying them. Some examples of these solutions already exist using tags, phones, and more.

Even though these solutions exist, all of them require reliability on a user held item. While this doesn't seem to be a problem, when deploying such a solution in a more public environment, it becomes one since we can't expect to oblige a user to carry an item while in a public environment.

Localization applications are common nowadays using technologies such as GPS. Although GPS is a good technology to for outdoors localization with does not work well indoors (Sugano et al., 2006).

Using algorithms based on the already existing location ones, it is possible to have a system which can locate objects indoors with the help of ultrasounds. These located objects can try to be refined into organic or not, by using the help of infrared sensors.

But to have such a localization system an infrastructure must exist beforehand which sometimes ought to be expensive which isn't appealing to implement. Also note that if such an infrastructure is implemented using unknown or unused technologies it might happen that such infrastructure reveals to be hard to manage and eventually might be abandoned.

This paper develops the idea of having a system self-reliable, which does not rely on external sources to be aware of a target's position. We developed our system to be a fully fledged distributed system with all the components it should have, the WSN itself which gathers the data, an Application programming interface (API) service for the end-user to interact and gather data, and a database to store data. We also aim to focus on wireless part on the system and developed an event based pseudo-scheduler in order to guarantee this type of objective.

2 BACKGROUND

First, we present some necessary background information for our system design.

2.1 Indoor positioning technologies

Indoor positioning systems normally use an hybrid solution using two technologies to have a correct position. These two normally complement each other and correct each other as well. It is very common that one of these technologies requires the target of positioning to carry some device which will then assist in the positioning of the target.

However it is possible to locate a target with only one of this positions getting the discomfort out of the target from carrying an external object solemnly for that purpose. This types of systems will however be less accurate than the hybrid since they rely only on one technology.

2.2 Communication within the WSN

In terms of communication within the WSN we focused mainly on Bluetooth, Wi-Fi, and Zigbee, since these are the most used technologies in WSN. Wi-Fi technology is a really high performance technology in terms of communication. One of the big problems of this technology is the energy consumption, but with a descent sized energy storage, it is possible to have the node functioning for a few days without the necessity of recharging the battery. Additionally, with the high data transmission ratio it is possible to send large loads of information without significantly affecting the overall consumption. Bluetooth was position as a short distance substitute alternative interfering with 802.11b (Chiasserini and Rao, 2001). Recently Bluetooth concept has evolved to a protocol suitable to supporting more complex ad-hoc networks with specific requirements, especially WSN. ZigBee and XBee wireless modules are simple and robust in their protocol. ZigBee is designed for use in personal networks of low bandwidth, low power consumption, low cost and high level of security (K. Sohraby, 2007), but lately, with the latest evolutions in the field, it has been used in industrial use as well.

3 DESIGN AND IMPLEMENTATION

Figure 3 presents an high-level architecture of the BAT system. This system is based onto three big components. The first component if the API, this component will act an entry point for the end-user to request information from the WSN. After receiving results from the WSN the API will then apply the trilateration algorithm onto the results which will result into positions and those positions will be returned to the

requesting user. Finally the API server will store all the gathered data into the database server which sole purpose is to store data for later usage. The final and most important component is the WSN, this is composed of two sub-components which are master node and minion nodes, following the master-minion architecture. The master nodes are in charge of receiving the requests from the API server and forwarding and orchestrating the minion nodes to execute the request accordingly and orderly. The master node is also responsible of making sure that the WSN is operating normally, making sure that for example nodes which die get removed from the system.

3.1 API design

The API server will serve as a front end to receive requests from the end-user. This API support multiple functions.

Read operation which has no parameters and return a Javascript object notation (JSON) object containing the position obtained, an identifier for the calculation and a list containing the measurements of all the nodes. Each measurement includes, an identifier for the node, and the actual measurement of the node.

Identify operation which has a parameter id which is the identifier of the node to physically identify. Afterwards it returns a JSON with successful or failed (as all responses do in our system.

List measurements operation which lists all the measurements taken so far. This operation takes no parameters and the response includes a list of measurements of the type explained above.

List positions operation which list all the positions calculated so far. This operation takes no parameters and the response includes a list of positions of the type explained above. A list of these method and specifications is presented in table 1.

Method	HTTP method	endpoint	parameters	response
Read	GET	/read	None	`{status: status type, pos_x: x_calculated, pos_y: y_calculated, measurements: [ id: id_of_node, distance_calculated: distance_in_cm ] }`
Identify	GET	/identify	id: id of the node	`{status:status_type}`
List measurements	GET	/listm	None	`{status:status type, measurements: [ id: id_of_node, distance_calculated: distance_in_cm ] }`
List positions	GET	/listp	None	`{status:status type, positions: [ calc_id: id for calc, pos_x: x_calculated, pos_y: y_calculated, time: time of calc ] }`

Table 1: Comparison of position technologies

3.2 Database design

For our database we have a simple schema.

We need a table which stores measurements. This table should have a column to store id the id of the node, a column to store the distance calculated and a table to store the time of when the measurement was done.

We need a second table for position calculations, this table should contain a column to store the id of the calculation, a column to store the x position calculated, a column to store the y position calculated, and a column to store the time of the calculation. We should then have a foreign key from the measurements table to map into a position calculation so that we can backtrack on which measurements generated which positions. An entity relation diagram can be seen representing such schema in figure 1.

Figure 1: Database schema for BAT.

3.3 WSN design

In order for this WSN to work properly, some custom protocols were designed and implemented, Dynamic ID Protocol (DIDP), read, and identify. The identify protocol is a simple one where the API gets a request to an identify endpoint and retrieves an id argument and the WSN will then proceed by physically iden-

tifying the node with such id by blinking a light on it.

The DIDP protocol is the starting point for the WSN. When the master node connects it will be listening for connections on the DIDP port via UDP. After a connection is received the master node will then respond with an identifier for the requesting node and will wait for the acknowledge. If such acknowledge does not arrive the master cancels the protocol and rolls back the node registration. If the acknowledge does arrive then the master completes the protocol successfully and the node is registered. For easier understanding figure 2 explains visually how this protocol works.

The read protocol is the main protocol for positioning. The API server will receive a request to read positions. It then proceeds to forward it properly to the master node which will then start emitting a broadcast into the read port where the minion nodes are listening. The minion nodes after receiving the requests will then start a pulseIn operation to retrieve data and return it to the master. The master waits for all the responses to come, however it will timeout after two seconds since measurements after that time will become irrelevant.

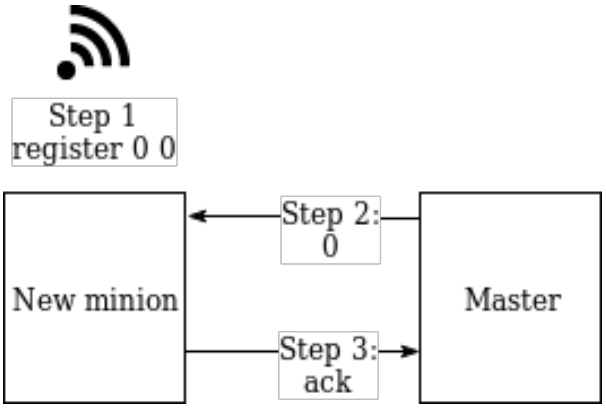

Figure 2: DIDP in a high level view.

In order to make the system resilient to the wireless part of a WSN we developed a pseudo-scheduler in order to keep important processes always ready to execute. The process we priority above all is the Over the air (OTA) which allows us to upload code without having to connect any wires into the micro-controllers, making it easier to manage the system. The pseudo-scheduler implemented has a basic scheduling functionality and supports only one priority thread (other than the OTA which is hard coded on purpose) which makes it inflexible to any other new changes. The objective of the scheduler is not to make the system faster but rather make the system more resilient and manageable for the maintainers.

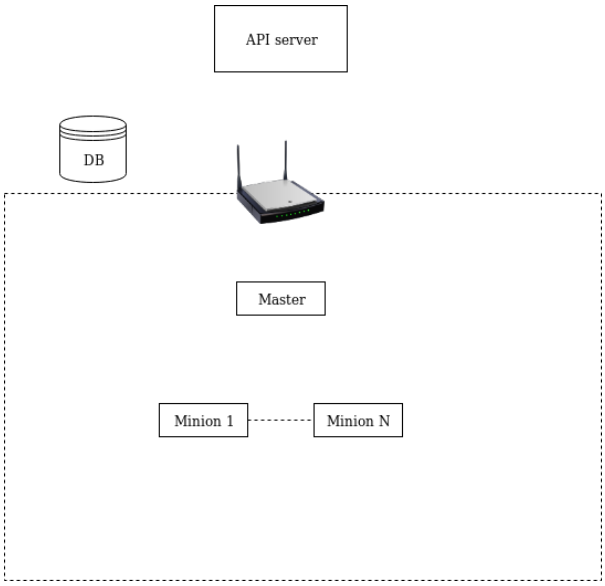

Figure 3: BAT high level design.

3.4 Prototype implementation

We implemented a BAT prototype using the ESP8266 micro-controller for both master and minion. For the minion nodes we also used the HC-SR04 ultra-sound sensor which allows us to both emit and receive ultra-sounds making it possible for the system to retrieve measures from the real world. As for the API server it was written using Go programming language due to its efficiency in operation and easiness on making web applications. The database server was implemented using a SQL database implementation MariaDB to provide persistent yet fast storage. Both the API server and the database server run in a Linux based distribution.

4 EVALUATION

To study the performance and reliability of the BAT system we measure the overhead of our scheduler in the system for the most relevant operation which is the read operation. We then evaluate the precision and accuracy of our trilateration algorithm.

The API server machine complies of a machine with an Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz CPU and 16 GB of RAM DDR4, on top of that it runs Gentoo Linux with default/linux/amd64/17.0 profile. Our API server will also be proxied by an NGinx high performance web server. The Golang version used go1.13.1 linux/amd64.

The database server machine complies of a raspberry pi 3B with an armv7l Cortex-A53 CPU and 1

Distance measured	without scheduler	with scheduler
4	429	430
5	408	410
10	1016	2005
100	637	637
1000	3437	3437
3000	1334	1503
3500	1358	1400

Table 2: Evaluation of read operation with and without scheduler.

GB of RAM, on top of that it runs Gentoo Linux with default/linux/arm/17.0/armv7a profile. The database service is a mariaDB Ver 15.1 Distrib 10.2.22-MariaDB, for Linux (armv7l) using readline 7.0.

The WSN nodes we use the ESP8266 micro-controller which is a low cost Wi-Fi microchip with full TCP/IP and micro-controller capabilities. This controller has a L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz, as a CPU 64 KB RAM and 96 KB of data RAM, and an Integrated TR switch, balun, LNA, power amplifier and matching network.

For the gateway between the WSN and external network we use a tplink tl-wn821n running default tplink firmware and without any specific type of configuration.

Table 2 shows the measurements obtained with and without scheduler in a system with only one master and one minion node. We can check that with the scheduler we have a bit of an overhead on the requests but not significant taking into account the extra benefits the scheduler provides. We can also check that even though the scheduler starts off with a wrong prediction it eventually stabilizes into a right prediction making the run time of the process be the same with and without scheduler, giving us the extra reliability on the system.

To study the accuracy and precision of our trilateration algorithm we prepared a lab environment as present with four slave nodes one in each corner of the lab environment. We start by booting up our system with the correct coordinates. We then put an object in a fixed coordinate and start a read operation. After the read operation as finished we check our results and put them in a scatter graph as presented in figure 4. As expected the results of the system are good but not as accurate as a system with other location technologies would be. We can also check from the scatter graph that the precision of the system is not the best do to the possible external changes that affect the system and, even more, there are some factors that also affect the system such as the way the sensors are positioned.

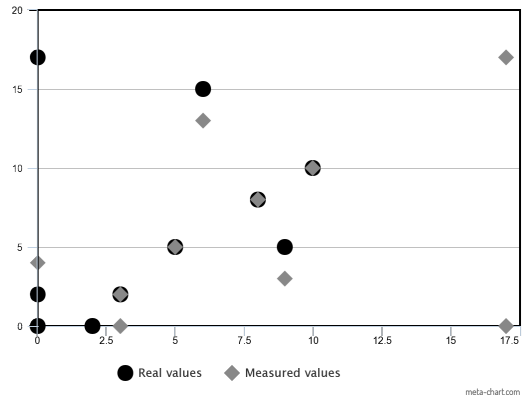

Figure 4: Scatter chart of the measures taken and the real values.

Figure 4 presents a scatter graph containing the real and the calculated values so we can easily visualize the accuracy and precision of the BAT system. We could not find any system that uses solemnly the ultra-sounds to position the target, however comparing with hybrid systems such as the one presented in (Holm, 2009) and the one presented (Hu et al., 2013), our results are worse both in accuracy, but again as was expected since hybrid systems have correcting points for a greater accuracy and use the best of two technologies to achieve this.

5 CONCLUSIONS

This paper presented BAT, an indoor positioning system. A fully fledged system complaint of a WSN, a database, and an API. Being also an easy to maintain system due to its scheduler made specially to keep operations such as OTA running permanently and in a reliable way. We also show that our system is not as performant as it should be expected but it is more maintainable than expected.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422.

Amundson, I. and Koutsoukos, X. D. (2009). A survey on localization for mobile wireless sensor networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5801 LNCS:235–254.

Buzně, E. and Cernea, D. (1991). Atitudine terapeutică în neuropatia optică prin alcool metilic. *Oftalmologia (Bucharest, Romania : 1990)*, 35(1):39–42.

Chiasserini, C. F. and Rao, R. R. (2001). A comparison between collaborative and non-collaborative coexistence mechanisms for interference mitigation in ism bands. In *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No. 01CH37202)*, volume 3, pages 2187–2191. IEEE.

Filonenko, V., Cullen, C., and Carswell, J. (2013). Indoor Positioning for Smartphones Using Asynchronous Ultrasound Trilateration. *ISPRS International Journal of Geo-Information*, 2(3):598–620.

Holm, S. (2009). Hybrid ultrasound-RFID indoor positioning: Combining the best of both worlds. In *2009 IEEE International Conference on RFID, RFID 2009*, pages 155–162.

Hu, W. Y., Lu, J. L., Jiang, S., Shu, W., and Wu, M. Y. (2013). WiBEST: A hybrid personal indoor positioning system. *IEEE Wireless Communications and Networking Conference, WCNC*, pages 2149–2154.

K. Sohraby, D. Minoli, T. Z. (2007). Wireless sensor networks. pages 79–88.

Sugano, M., Kawazoe, T., Ohta, Y., and Murata, M. (2006). Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. *Target*, 538:050.

Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330.

Zarocostas, J. (2008). WHO returns to Iraq to find major problems among 2.8 million internally displaced Iraqis. *BMJ (Clinical research ed.)*, 337(1):13–32.

Zhang, D., Xia, F., Yang, Z., Yao, L., and Zhao, W. (2010). Localization technologies for indoor human tracking. *2010 5th International Conference on Future Information Technology, FutureTech 2010 - Proceedings*, (60903153).