

# A hybrid indoor localization solution using a generic architectural framework for sparse distributed wireless sensor networks

Tom Van Haute\*, Jen Rossey, Pieter Becue, Eli De Poorter, Ingrid Moerman, Piet Demeester  
Ghent University - iMinds, Department of Information Technology (INTEC), Belgium

\* tom.vanhaute@intec.ugent.be, Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium, +32 9 331 49 46

**Abstract**—Indoor localization and navigation using wireless sensor networks is still a big challenge if expensive sensor nodes are not involved. Previous research has shown that in a sparse distributed sensor network the error distance is way too high. Even room accuracy can not be guaranteed.

In this paper, an easy-to-use generic positioning framework is proposed, which allows users to plug in a single or multiple positioning algorithms. We illustrate the usability of the framework by discussing a new hybrid positioning solution. The combination of a weighted (range-based) and proximity (range-free) algorithm is made. Both solutions separately have an average error distance of 13.5m and 2.5m respectively. The latter result is quite accurate due to the fact that our testbeds are not sparse distributed. Our hybrid algorithm has an average error distance of 2.66m only using a selected set of nodes, simulating a sparse distributed sensor network. All our experiments have been executed in the iMinds testbed: namely at “de Zuiderpoort”. These algorithms are also deployed in two real-life environments: “De Vooruit” and “De Vijvers”.

## I. INTRODUCTION

Combining wireless sensor network nodes with the upcoming trend of smartphones creates a totally new range of possibilities. Normally, wireless sensor networks are used to monitor a certain environment and measure e.g. the temperature and humidity. But also tracking of persons and equipment can be done by sensor networks. GPS [1] is the traditional way of tracking people or vehicles outdoor, however this does not work properly indoor because line of sight (LOS) is required to receive the GPS signals.

Sensor nodes inside buildings can fix this issue, however other factors have to be taken into account: interference, infrastructure, the amount of sensor nodes that is required, energy consumption,... It will always be a trade-off between cost and accuracy. Further, a myriad of positioning algorithms have been developed in the last few years. A standalone solution generally does not offer sufficient accuracy in different environments (indoor/outdoor, different type of buildings,...). In this paper however, we will try to find a solution with an acceptable accuracy when only a sparse distributed sensor network is available. Our algorithm described in this work is a combination of two already existing algorithms. Each belonging to a different subdivision, namely range-free and range-based. Both solutions show too many defects in thinner environments. Combining them results in a noticeable improvement. In this way, room accuracy can be guaranteed.

The rest of the paper is organized as follows. In section 2, the already existing algorithms are clarified. Section 3 describes our generic architecture framework. The hybrid algorithm build in this framework whereby the two previous are combined is discussed in section 4. Section 5 handles about our different testbeds. The experiments with their results are summarized in section 6. Finally, some conclusions can be made. These are, together with the future work, clarified in section 7.

## II. RELATED WORK

In this section, we conclude the work that is essential to comprehend our hybrid solution. Localization algorithms can be subdivided in two different categories. The first category is called the “range-based”-algorithms. In order to calculate a position pertaining to multiple fixed nodes, a distance measurement is essential. Then, on the base of this distance, a position can be determined by means of trilateration. Trilateration is a method to find the intersection of three circles whose center and radius are known. There are many different ways to measure the distance. The most familiar techniques are Received Signal Strength Indication (RSSI), Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Angle of Arrival (AoA).

The second category, “range-free”-algorithms, does not require a distance measurement to calculate the position of a sensor node. They are based on the information of the connection. If two sensor nodes can connect to each other, than the maximum distance between them is the maximum transmission range. Thus the position of the mobile node can be estimated with this information. This is a very simple and cheap technique. Moreover the accuracy will depend on the density of the wireless sensor network. Centroid, triangle elimination and proximity are common range-free algorithms.

The hybrid solution uses both techniques. A combination of a range-based and a range-free algorithm is made. In the following two sections, both algorithms will be explained more in detail.

### A. Range-based: weighted

The first one is a range-based solution described by Tareq Ali Alhmiedat et al. in [2]. The proposed algorithm is based on weighted RSSI values. The main idea of RSSI is that the transmission power  $P_T$  directly affect the received power  $P_R$

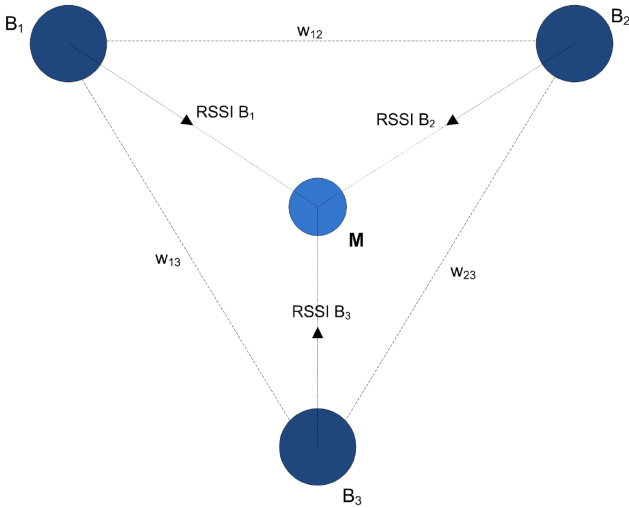


Fig. 1: Weighted algorithm: schema

of a signal. Via the Friis transmission equation, also defined in [2], the linear relationship becomes clear:

$$P_R = P_T * G_T * G_R \left( \frac{\lambda}{4\pi d} \right)^2 \quad (1)$$

where  $G_T$ ,  $G_R$  are gain of transmitter and receiver respectively.  $\lambda$  is the wavelength of the signal and  $d$  is the distance between sender and receiver. The received signal strength indicator (RSSI) can be defined as the ratio of the received power to the reference power  $P_{Ref}$ .

$$RSSI = 10 * \log \frac{P_R}{P_{Ref}} \quad (2)$$

Each RSSI value can be matched with a certain distance. The proposed algorithm in [2] not only uses the RSSI-values to measure the distance between a fixed and mobile node, but also the distance between the fixed nodes mutually is measured. These values function as weight factors for the distance calculation between the fixed and mobile node. These weight factors are shown in Figure 1 as  $w_{12}$ ,  $w_{13}$  and  $w_{23}$ . The distance from  $M$  to, for example,  $B_1$  can be calculated as follows:

$$Distance(M, B_1) = \frac{RSSI B_1 * w_{12} + RSSI B_1 * w_{13}}{2} \quad (3)$$

Their results prove that these weight factors add value to the accuracy. A drawback of the RSSI technique is that these measurements are very sensitive to the environment and potential changes in it. The relationship between the distance and RSSI depends on the room. For example, in a long corridor, the fixed nodes their signals will have a greater range because they reverberate through the long walls. In this way, completely different results can be obtained.

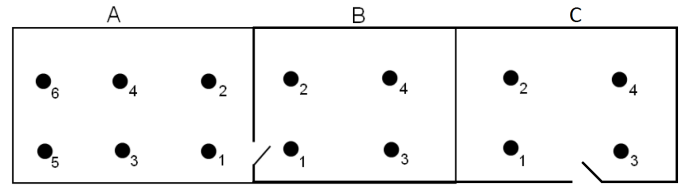


Fig. 2: Three neighboring offices

### B. Range-free: proximity

In contrast to the previous category, range-free algorithms do not take RSSI-values into account. If a mobile sensor node has a range of 10 meters, than a fixed node can only receive his messages if the mobile node is maximum 10 meters away. This is the only information that is used to calculate the position of a mobile node. This technique is used by J. Wyffels et al. in [3]. A proximity-based algorithm is used to localize the patients and the nurses in a healthcare environment. Important here is that the transmission power is well configured. If the power is too low, the mobile node could be out of range between two fixed nodes. And also vice versa if the power is too high, too many fixed nodes will receive the beacon and a wrong estimation could be made.

The latter problem can be solved by using a centroid algorithm [4]. This is only useful if there is a set of fixed nodes with an overlapping coverage area. The beacon of the mobile node is received by multiple fixed nodes. In order to determine the position, the centroid of all the receiving fixed nodes is calculated (Eq. 4).

$$[x_M, y_M] = \left[ \frac{\sum_{n=0}^{k-1} x_n}{k}, \frac{\sum_{n=0}^{k-1} y_n}{k} \right] \quad (4)$$

Normally would this algorithm give a 100% guarantee that room-accuracy is ensured. However, experiments have shown that this is not always the case. If the walls are small enough and/or not made of concrete, signals can go trough and a fixed node in a different room can catch up the beacon. In order to prevent a wrong location estimation, some extra logic can be implemented in the algorithm.

To implement the extra logic, some extra information is necessary as well. Suppose we have the exact coordinates of all the walls, doors and nodes inside a building. Knowing that every beacon has an index number, the direct path could be checked between the two fixed nodes who received the consecutive beacons. If the mobile node goes from one room to another, without using a door, then the last beacon can be dismissed. For example (Fig. 2) when node  $A_2$  receives a beacon and the next beacon is received by node  $B_2$ . It is impossible to move directly from  $A_2$  to  $B_2$  without passing nodes  $A_1$  and  $B_1$ . So the message that was received by beacon  $B_2$  will be rejected.

With this optimization room-accuracy can be guaranteed. Still, this solution has the drawback that a lot of fixed sensor nodes are necessary to retrieve good results. If the network is sparse distributed, then the algorithm would not work properly.

### III. POSITIONING FRAMEWORK

The framework is developed in Java and consists of three parts: the positioning server, the web server and the client application (Fig. 3).

The positioning server has two functional blocks. The interconnection gateway is responsible for the retrieval of positioning information gathered by the network infrastructure or mobile unit that is being located. The interconnection gateway further incorporates an abstraction layer which hides the underlying technology (ZigBee, Wi-Fi, Bluetooth, ...) from the positioning server. In Figure 3, two different approaches for positioning in wireless sensor networks are shown. On the left side, a mobile device broadcasts positioning beacons and the sink node of the WSN forwards the beacons to the interconnection gateway. On the right side, the infrastructure nodes broadcast beacons and the mobile unit collects and forwards the beacons to the interconnection gateway. The interconnection gateway further passes the positioning information to the position calculator, which consists of pluggable positioning algorithms. Multiple positioning algorithms can be active at the same time. A reasoner is used to select the algorithm giving the most accurate position or to intelligently combine the results of multiple algorithms into a more accurate (hybrid) position. Map info can also be taken into account when calculating the position.

The web server can poll the positioning server for the user's position. And the client application can either run on a smartphone or a central monitoring station. The client communicates with the web server through e.g. Wi-Fi or Ethernet.

Some advantages of the framework:

- Existing smartphone applications can use position information by implementing a simple interface allowing the application to request a user's position from the web server.
- Conversion of relative coordinates to GPS notation is possible. This implies that client applications developed to work outdoor (GPS), can easily use this framework.
- The user of the client application can pinpoint his correct location on the floor plan (for testing purposes). The application then calculates the difference between the estimated and the real position, thus allowing the user to evaluate the algorithm.

### IV. HYBRID ALGORITHM

Having this framework described above, designing a hybrid solution is very efficient. The reasoner allows the position calculator to combine the results of different algorithms and other available information. In the hybrid solution the reasoner has two choices: if the mobile node is in range of a fixed node we use the result of the proximity algorithm. If no fixed node can hear the proximity node, the reasoner will decide to use the weighted RSSI algorithm, where the mobile unit has a wider range.

The biggest problem of the stand-alone weighted algorithm, is the selection of the nodes. An ideal situation would be that

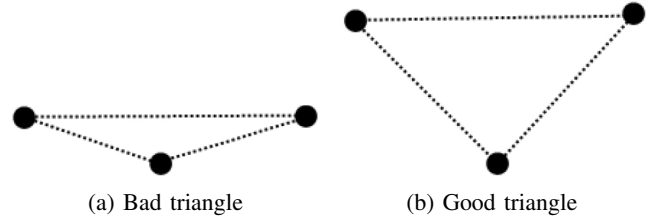


Fig. 4: Node selection

the proximity node would be surrounded by the closest fixed nodes. As discussed in subsection II-A, selecting the closest nodes is not possible if only the RSSI values are available. The proximity algorithm can give extra information whereby finding the closest nodes can be realized. Hence, the node selection can be optimized using the latest information of the proximity algorithm. In that way, the first node of the triangle is determined. In order to have a good coverage of the area, the two other nodes must be well selected. If the angles of the triangle are too sharp (Fig. 4a) than the weighted algorithm will not function properly. In certain situations, the two last nodes will have to be reselected until a good triangle (Fig. 4b) is founded.

**Data:** Three circles of each fixed node

**Result:** Position of the mobile node

```

if three circles do not intersect then
    while smallest circle does not intersect with the
        second smallest circle do
        | increase the smallest circle
    end
end
// Now at least two circles intersect
Calculate the intersection of the two smallest circles
position mobile node = intersection of the two smallest
circles closest with the biggest circle

```

#### Algorithm 1: Adapted trilateration

Once the three fixed nodes are selected, a distance measurement is the next step in the procedure. This is done the same way as the stand-alone weighted algorithm (Eq. 3) except for one thing. The RSSI values are slightly adapted because results of previous experiments have shown that the calculated distance was almost always too big. This adaption is estimated experimentally. After the distance calculation, the three circles can be created and trilateration can be applied. In perfect circumstances, the three circles will intersect in exactly one point. However, in practice this is never the case. Due to the environment and interference, the three circles will never intersect in one single point. Therefore, an adapted trilateration technique is shown in Algorithm 1.

Finally, if the reasoner has access to other input, such as information about walls, rooms, doors, we can use this to influence our position estimate.

### V. ENVIRONMENT DESCRIPTIONS

This positioning framework including the hybrid solutions has been tested in two wireless testbeds and also in two dif-

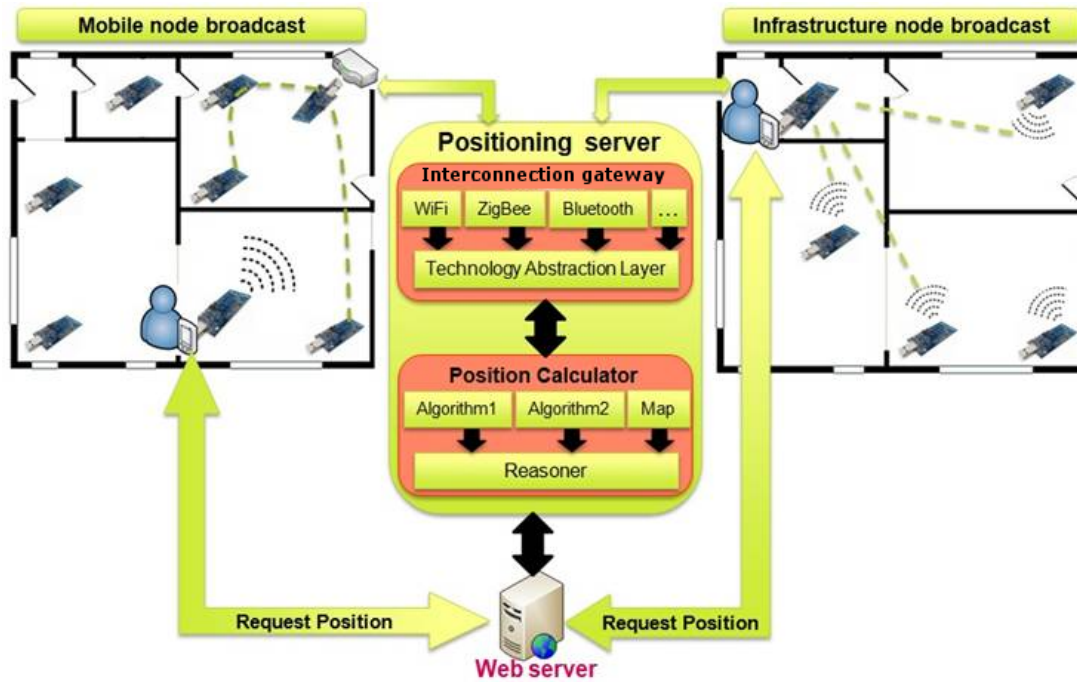


Fig. 3: Framework architecture

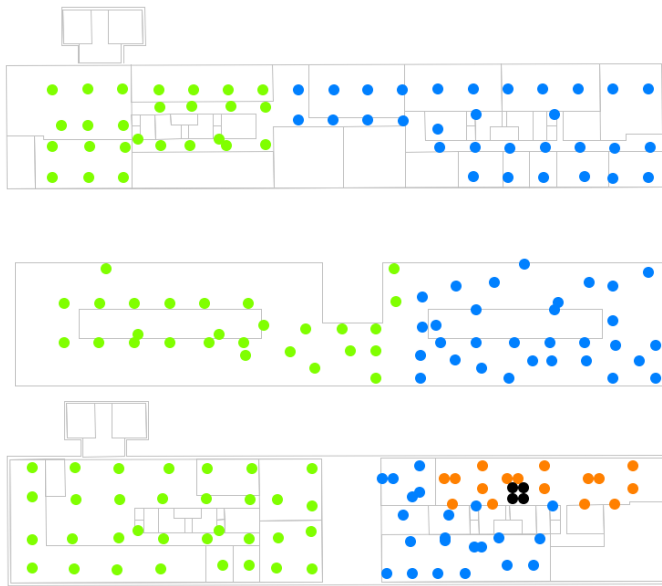


Fig. 5: w-iLab-t at the “Zuiderpoort”. 200 nodes on three floors (18m x 90m). The different colors indicate the nodes can be divided in groups for executing tests.

ferent real-life situations. Each environment will be explained further in detail in the next sections.

#### A. w-iLab-t at the “Zuiderpoort”

The w-iLab-t is an extensive facility that is introduced in detail by S. Bouckaert et al. in [5]. The infrastructure is distributed on three floors of the iMinds office in Ghent, Belgium (Fig. 5). The network consists of 200 nodes. Each node

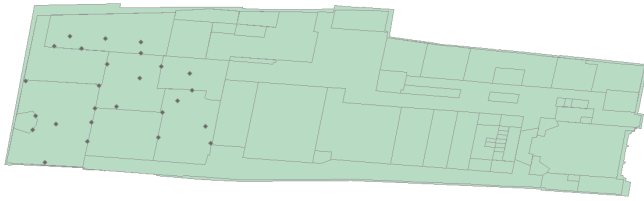
has (i) a Tmote Sky IEEE 802.15.4 mote, (ii) two Compex WLM54SAG 200mW AR5006XS 802.11a/b/g 54/108 Mbps miniPCI wireless cards and (iii) an environment emulator. The latter one is self-made and used for simulations: environment (e.g. temperature change), battery drop, user input, etc. These nodes are centrally managed for control and monitoring purposes and remote access by using an Intel x86 architecture (PC Engines Alix Boards).

#### B. “De Vooruit”

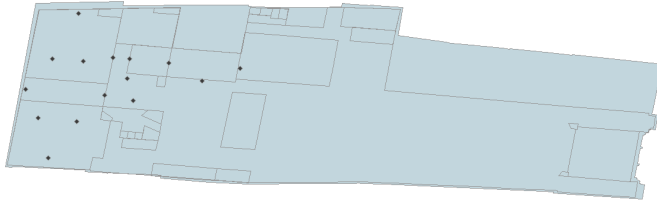
De Vooruit is an ancient building close to the historical center of Ghent [7]. In the past, this building was a place for the working class where they could eat, drink and enjoy culture at democratic prices. Since 1982 De Vooruit is recognized as a monument and nowadays it is still used to organize lectures, debates, concerts, parties, ... This location was a perfect use case to test the indoor localization solutions. Due to the fact that the building was recognized as a monument, it was not allowed to use a cabled network. In this situation, wireless sensor networks were the only solution to handle this problem. 50 nodes, distributed over four different floors (Fig. 6), were used to locate the mobile nodes worn by the visitors. In this use case, Sentilla JCreate nodes in combination with battery packs were used.

#### C. “De Vijvers”

As a second use case, the positioning was tested in a home for the elderly. The goal here was to track people with dementia that are not allowed to leave the home. When a person goes in a restricted zone, an alarm was sounded. The position of the person could then be seen on a map in the reception. In this building (Fig. 7), 25 Sentilla JCreate



(a) Fourth floor of De Vooruit



(b) Sixth floor of De Vooruit

Fig. 6: Floorplan of the fourth and sixth floor of De Vooruit, the diamonds represent the fixed nodes installed in the building.



Fig. 7: The southern part of “De Vijvers” where 25 nodes were attached in the central and eastern part of the building. Their positions are marked with red dots.

nodes were attached to obtain the required accuracy for this application.

## VI. RESULTS

In this section, we present the results of all the interesting measurements. First, the two algorithms are tested separately, followed by the results of the hybrid solution. All these measurements are done at De Zuiderpoort (Section V-A) on the third floor.

### A. Range-based: weighted

The results from [2] showed that the weighted RSSI-values give a more accurate position than the normal RSSI-values. For

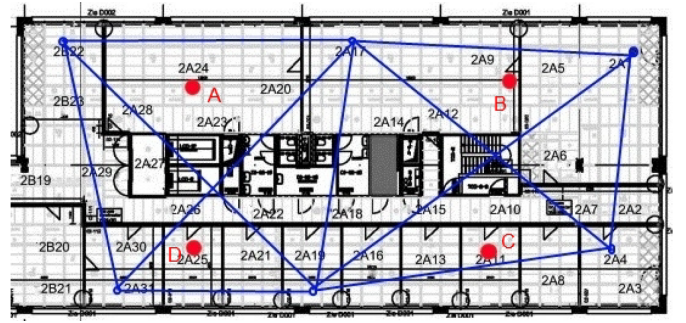


Fig. 8: Dividing the third floor in big triangles for the second test of the weighted algorithm.

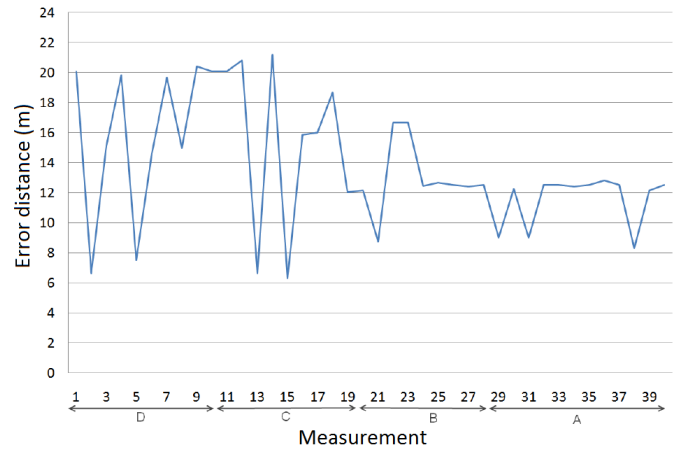


Fig. 9: Graphical overview of the results of the weighted algorithm using a sparse distribution of the fixed nodes

those reasons, only the results of the weighted algorithm will be shown.

Several tests have been executed. First, all available nodes on the third floor were used. This, however, gave very poor results so they are not published in this work. Some measurements had an error distance of more than 20 meters. An explanation for these large error distances is multipath fading of the nodes in the corridor. The setup of the second test is shown in Figure 8. The third floor is divided in big triangles (marked with blue lines) to calculate the position of *A*, *B*, *C* and *D* (marked with a red dot). The results of these measurements can be found in Figure 9. For each location, ten measurements are executed. The smallest error distance is 6.3m, the biggest is 21.2m with an average error distance of 13.8m. These results are not acceptable because room accuracy cannot be guaranteed. The large error distance is due to the fact that a high transmission power was necessary to communicate through the concrete walls in the center of the building. The concrete walls has a strong influence on the RSSI measurements. For those reasons, a third test was implemented that avoids the concrete walls.

The triangles of the third test can be found in Figure 10. In this way, the signals do not need to go through the concrete walls so a lower tx-power can be used. The results of this



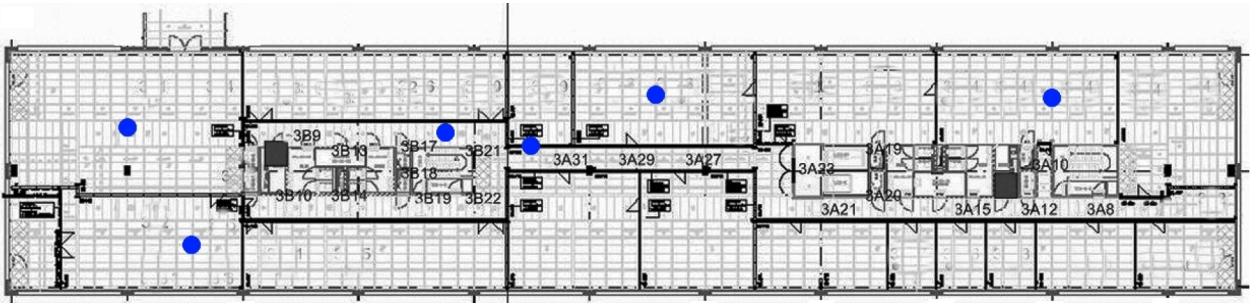


Fig. 12: Positions of the mobile node for testing the hybrid localization solution

TABLE I: Summary of the experimental results

LOCALIZATION SOLUTION	ERROR DISTANCE (IN METER)		
	MINIMUM	MAXIMUM	AVERAGE
Weighted algorithm ( <i>big triangles</i> )	6.3	21.2	13.8
Weighted algorithm ( <i>small triangles</i> )	0.5	8.2	4.8
Proximity algorithm	0	2.5	-
Hybrid algorithm ( <i>all nodes</i> )	0.49	8.5	3.28
Hybrid algorithm ( <i>filtered nodes</i> )	0.49	8.5	2.66

distances are collected in one organized table. It becomes clear that the hybrid solution has an improvement (if you compare the average error distances).

## VII. CONCLUSION AND FUTURE WORK

This paper presents a hybrid indoor localization solution designed to achieve room accuracy using sparse distributed sensor networks. Hereby a positioning framework is developed to accomplish a hybrid solution, based on two existing solutions.

The positioning framework consists of two functional blocks: the interconnection gateway and the position calculator. The interconnection gateway gathers all the necessary data from the fixed and mobile node in order to calculate the position. This data can come from any kind of hardware device/technology. The position calculator contains all the different localization algorithms that calculate the position of the mobile node using the data from the interconnection gateway. This calculator includes also a reasoner that decides which algorithm calculates the most accurate position at a certain moment, it can also combine multiple algorithms to improve the accuracy even more.

The hybrid solution is based on a range-based and range-free algorithm. The former is a category of techniques that requires distances measurements in order to calculate the position of a mobile node. These distance measurements can be done in different ways. In this paper, the range-based “weighted” algorithm is proposed. It uses RSSI measurements to calculate the distance between the nodes. The higher the value, the shorter is the distance between the nodes. Innovative here is that RSSI measurements are also used to calculate the distance between the fixed nodes mutually. Using this extra information, a *weighted* distance calculation can be done using triangulation.

The range-free solution “proximity” does not require these distance calculations, the localization is only based on the

information of the connection. This means that the reception range of a fixed node is as well as the maximum error distance. However, an extra optimization is possible, if multiple fixed nodes receive a beacon, then the centroid of all the fixed nodes can be calculated and be assumed as the point closest to the mobile device.

Both algorithms show issues in sparse distributed sensor networks. The accuracy of the weighted algorithm is far from acceptable because it is not easy to determine the correct triangle for the calculation and the proximity solution is completely depended on the density of the fixed nodes. Therefore, combining both algorithms can resolve the biggest issues of both solutions. First a proximity beacon is received by a fixed node, this is the first corner of the triangle. Then the other two corners are determined in order to get a good triangle.

In the results, it became clear that the improvement of the hybrid solution is significantly. The average error distance dropped from 13.8m/4.8m to 3.28m/2.66m. Still, some future work can be done. First, the issue with the transmission power must be tackled. Further, comparative tests using WiFi or other technologies are in progress.

## ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no 317989 (STREP EVARILOS).

## REFERENCES

- [1] Bulusu, N.; Heidemann, J.; Estrin, D. ‘GPS-less Low Cost Outdoor Localization For Very Small Devices’
- [2] Alhmiedat, T.A. and Yang, S-H. (2008) ‘A ZigBee-based mobile tracking system through wireless sensor networks’, *Int. J. Advanced Mechatronic Systems*, Vol. 1, No. 1, pp.63-70.
- [3] Wyffels, J.; Goemaere, J.-P.; Verhoeve, P.; Crombez, P.; Nauwelaers, B. and De Strycker, L. (2012) ‘A novel indoor localization system for healthcare environments’, pp.1-6.
- [4] Qiu, T.; Zhou, Y.; Xia, F.; Jin, N.; Feng, L. (2012) ‘A localization strategy based on n-times trilateral centroid with weight’, *Int. J. Communication Systems*, pp.1160-1177.
- [5] Bouckaert, S.; Vandenbergh, W.; Jooris, B.; Moerman, I.; Demeester, P. ‘The w-iLab.t testbed’.
- [6] Bouckaert, S.; Becue, P.; Vermeulen, B.; Jooris, B.; Moerman, I.; Demeester, P. ‘Federating wired and wireless test facilities through Emulab and OMF: the iLab.t use case’.
- [7] De Vooruit, the monument. Online on 29 March 2013 (in Dutch) - <http://vooruit.be/nl/information/detail/36/Monument>