A Logic Range-free Algorithm for Localization in Wireless Sensor Networks

Balzano Walter
Dipa. Ing. Elettrica e Tecnologie dell'Informazione
Univeristà di Napoli, Federico II
Napoli, Italy
Email: walter.balzano@gmail.com

Stranieri Silvia
Università di Napoli, Federico II
Napoli, Italy
Email: silviastranieri1047@gmail.com

***Abstract*—Nowadays, localization is a very important research problem in the context of Wireless Sensor Networks (WSNs). These networks are made by nodes connected to each other wireless and able to collect data: all the retrieved information can be then held by a central unit, a more performance machine, which can in turn collect and elaborate all the data. The main contribution of this work is in an innovative way to recognize the position of a point in a certain closed environment, by exploiting the RSSI (Received Signal Strength Indicator) and a logic approach: for this purpose, Prolog has been used in order to describe an intuitive non-greedy algorithm and an appropriate simulation program able to make an estimation of a point localization, providing the global optimum. To this aim, the placement of hub in the interesting area is crucial and, in particular, when this area presents some obstacles which can alter the transmission signal. The expressive power of Prolog and the way the its logic engine works made this programming language suitable for our purpose: in fact, the backtracking strategy opportunely reflects the way the relative positioning of hubs is performed with the aim of improving the cover of a specific indoor area at each step, which is a very important application of the localization problem.**

Key-words: Wireless Sensor Networks, localization, coverage, Prolog

I. Introduction

Wireless Sensor Networks constitute a spread field of research of the last years, due to the potential of that system which is made by nodes self-organized, broadcasting information and data all over the net. Many proposals have been made to improve the communication between the nodes, to select the best routing protocol possible, and to locate nodes inside the net. The localization is a crucial point in this issue, since a fast and efficient positioning technique can also provide a way to optimally cover an area, in such a way that a minimum signal is guaranteed to any point.

The hub distribution in an outdoor environment does not create any trouble, due to the absence of any kind of obstacle. In a closed area, instead, the presence of walls, doors, or any other impediment can alter the power of the transmission signal, and this is something we want to deal with in order to guarantee an optimal distribution of hub in indoor environment too.

The idea is to design and analyze an algorithm which preventively creates a mapping of hub inside the interesting area, then applies some localization techniques in order to estimate the position of a point in that area. The approach is the logic one, since the way the algorithm works is based on backtracking: whenever a hub is added into the environment, its total mapping is questioned and the position of each hub is computed again in order to optimize some metric on the signal (such as average or variance in the whole space). This approach provides a non-greedy algorithm whose solution is guaranteed to be the global optimum, rather than the local one. The algorithm presents two possible ways to start the mapping: (*i*) by using some fixed hub, whose position is known and uneditable for some logistic reasons; (*ii*) without any a priori fixed hub.

In this work, we consider three different kinds of localization techniques: range based, angle based, and range free, focusing our approach on the last one. Moreover, before performing the actual localization, we use a simulation program to generate the map of an environment with some obstacles, taking into account three possible nodes distribution: casual, geometric, and signal based.

The work is organized as follows: in section II, we give an overview of the actual state of the art, analyzing some works and proposes about localization in Wireless Sensor Networks, and strategies that use RSS measurements to improve some known localization techniques; in section III, we briefly introduce the most significant localization approaches, then we focus on our propose, providing a simulation program first, and a research algorithm next in order to estimate the position of any point on a network, with a logic approach; in section IV, we provide an example of simulation and we apply some analysis on the proposed technique, highlighting how the precision in the estimation can vary according to some values; finally, in section V, there are conclusions and future hints for the growth of this important research field.

II. Related Work

Wireless Sensor Networks are, nowadays, one of the most studied research topics. The development of such networks was initially born for military purposes, while now, as explained in [2], there is a bunch of applications of these nets: environment and structures monitoring, traffic management, surveillance, and many others application fields. Actually, this is the reason why many studies are made about this topic and all the related issues, such as localization of nodes in such a network and signal distribution. An important indicator which is largely used in Wireless Sensor Networks for localization purposes is the

DOI reference number: 10.18293/DMSVIVA2018-008

RSSI (Received Signal Strength Indicator). This indicator provides useful information about the signal power for any retrieved hub in the environment. For instance, in [6] RSSI is exploited in traffic control field in order to estimate the positioning of vehicles. They state that Global Positioning System does not always guarantee the accuracy needed in cooperative-vehicle-collision-warning systems, while the radio-based-ranging approach founded on RSSI improves the accuracy. Using the same approach, in [7] they propose a range-free algorithm based on RSSI comparisons, called Ring Overlapping. Each node uses overlapping rings in order to guess the possible area in which it lie: given an anchor node *A*, each ring is actually generated by comparing the RSSs received by a node from *A* and the ones received by other anchor nodes from *A*. Even in [20], they highlight the importance of positioning accuracy in vehicle-to-vehicle field.

A crucial variation point in localization algorithms in WSN is in the choice of using anchor nodes or not. In [3] is proposed an anchor-based localization approach: the main idea is that each anchor is aware of its position, because equipped with GPS, and it periodically shares its current location with the other nodes which are able, thanks to this information, to locate themselves. This approach tolerates the presence of obstacles and has the benefit of not requiring any hardware modification. Oppositely, in [12] they prefer an anchor-free approach, summarizing all the drawbacks of having fixed nodes in a network.

In our previous work, we focus on logic strategies in order to deal with many problems related to traffic control, such as in [15], sometimes integrating it with clustering techniques ([14], [16]), or Distance geometry problem, like in [17]: even in this work we use the logic approach (*i*) to facilitate the comprehension of the algorithm behavior, through elegant and compact code, and (*ii*) to exploit the expressiveness power of Prolog and its cut operator to prune useless computational paths. But, many other localization techniques are proposed in literature. In particular, in [4] they highlight three categories of localization approaches: (*i*) AOA (Angle of Arrival) represents the angle between the propagation direction and some reference direction (orientation) and it constitutes the information which is exchanged between nodes, so that their localization can be performed by using trilateration [8], (*ii*) Distance Related Measurements, and (*iii*) RSS (Received Signal Strength) profiling. Moreover, in [5] they propose an indoor localization approach, called EZ localization algorithm which estimates the positioning of 2D point in terms of absolute coordinates: latitude and longitude.

III. Background

In this section, we are going to describe our contribute to the localization problem in Wireless Sensor Network, by explaining how the range-free localization algorithm works, with the support of a simulation program, which generates an environment with nodes and obstacles where the signal power is represented and is used to locate a point. Before starting explaining our work in detail, (*i*) we briefly analyze the Wireless Sensor Networks and their topologies, (*ii*) we introduce the Received Signal Strength Indicator (RSSI), and (*iii*) we illustrate other localization techniques, such as the range-based and angle-based ones.

First, we introduce Wireless Sensor Networks, a system of nodes which exchanges data wireless. All this information can be possibly held and elaborated by a control center. As known, each net can have a particular topology, which characterizes the behavior of its component. In [1], they summarize essentially six kinds of network topologies:

1. Star topology: each node is connected to a single hub which filters any communication;
2. Ring topology: there isn't a leader, the information exchange follows one direction (the one of the ring);
3. Bus topology: there is a communication channel were all the information passes through;
4. Tree topology: hierarchical structure is the base of any communication;
5. Fully connected topology: each node is connected to any other node and this makes this topology suffer from NP-complexity;
6. Mesh topology: nodes have a regular distribution and each node communicates with its nearest neighbor.

Another important ingredient concerning localization is the Received Signal Strength Indicator (RSSI). This indicator provides the power of the received signal in a certain point and it has a strong relevance since not only it gives important knowledge for the purpose of localization, but it is also recognizable by any device on the market. For instance, *WirelessNetView* is an application which freely provides the percentage of the received signal by any retrieved hub.

We present the most famous approaches to estimate the position of a point in a Wireless Sensor Network. The initial classification we can introduce divides localization techniques in *anchor-based* and *anchor-free*: in the first approach, the network presents some special nodes, the *anchors*, which are aware of their position since they are equipped with a Global Positioning System, while all the other nodes, the *targets*, guess their location with respect to the anchors one; while someone actually prefers this kind of approach, such as in [13], some other authors have found some limitations in anchor-based algorithm, hence an anchor-free approach has been introduced. For instance, in [12] they suggest this kind of approach, since they indicate three reasons why the anchor-base algorithms are not the best choice: (*i*) there is a waste of time due to the manual insertion of anchor nodes; (*ii*) anchor-based algorithms are instable, since a small mistake in the anchors positioning may cause a huge mistake in the wireless sensor network final configuration; (*iii*) anchor-based algorithms are not scalable.

A. Range-based

Range-based algorithms can be classified into three main groups of approaches to the localization problem, which we are going to analyze. The first one is trilateration: this is a well known approach used by the Global Positioning System (GPS)

and it consists in the estimation of a point positioning by computing the intersection between four spheres, among which the Earth. Typically, the remaining three spheres are generated by satellites: for this reason, this technique is clearly optimal in outdoor environments, but it is not useful in case of indoor ones, or situations like "urban canyon", where points are not visible by satellites.

The second one is min-max: in order to locate a point P, each node creates a square around itself: the sides of this square are as far from the node as the distance between the node and P. This is made exploiting the values provided by RSSI. Eventually, the intersection between these squares is taken:

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \quad (1)$$

where x_i and y_i represent the coordinates of i-th point and d the distance between the i-th node and P. The estimation of the position of P is given by the center of the area obtained by the intersection, as explained in [9] and [10].

Finally, maximum likelihood: again, the distance between a node and the point P we want to locate is used, by exploiting the RSSI information. The aim here is to minimize the average quadratic error.

B. *Angle-based*

Angle-based localization technique provides a way to estimate the absolute position of a point in a given area: typically, the angular distance between nodes is computed and then used in order to estimate their position through trilateration.

Another possible application of this technique is the algorithm based on DoA (Direction of Arrival) shown in [11]. They explain how, given an antenna, the algorithm tries to estimate its location, by following three main steps:

1. Initialization: instantaneous angular estimate is computed;
2. Tracking: movements of the transmitter in the angular domain are traced in real-time;
3. Data mapping: angular estimates and other information about the height of the transmitter are used in order to project the antenna position over an indoor map.

IV. Our Approach

Our work focuses on *range-free* approach in order to provide a localization algorithm. This kind of localization technique uses some particular maps, called *fingerprinting radio-maps* where the signal power of each node is represented by its fingerprint (fingerprint-based techniques have been used in [19] and [21], too).

Since these maps are created through measurements of the retrieved signal in various points of the environment, the presence of obstacles has an impact on the signal power, as we can observe in Figure 1, where the variation of color intensity reflects the signal attenuation. Each device is represented by the black areas, and as the distance from it increases the strength of the signal decreases: this is expressed in a color variation from dark red to yellow. Moreover, we can observe that this variation is not regular nearby the obstacles (represented by the black lines): in fact, the presence of obstacles deforms the signal and lets the color turn into yellow more quickly (such as in the case of hub 2).

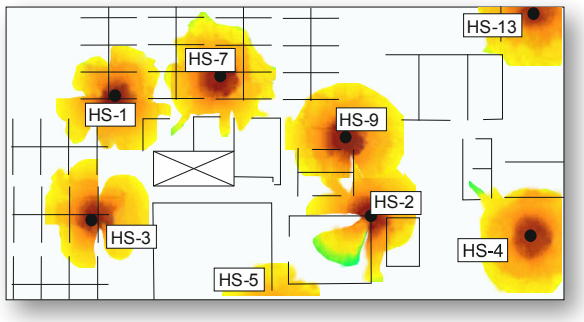

Figure 1: Generic fingerprinting radio-map with different nodes and variation of signal due to presence of obstacles

In this work, we propose an algorithm that uses this kind of maps and a simulation program to create them. For each point the power level is computed by using the inverse-square law:

$$P = \frac{P_M}{(x_i - x_n)^2 - (y_i - y_n)^2} \quad (2)$$

where (x_i, y_i) are the coordinates of one of the point, (x_n, y_n) are the coordinates of one possible point in the environment, P_M is the maximum signal for the node, and P is the computed signal power for that point with respect to that node. This law tells us that the signal power is inversely proportional to the square of the distance. Our simulation program works as follows: initially, it provides an environment in which we can put nodes and obstacles; then, it creates a list of points which represent the grid where we are going to simulate the retrieved power. For each obstacle, it picks a point and it generates for it the dimensions and an attenuation factor α which indicates how that obstacle influences the signal. In the end, a vector for each obstacle is obtained. Now, nodes have to be located and the program can do this by following three different node distributions:

- Random;
- Geometric;
- Signal-based.

Following a random distribution, nodes are placed randomly all over the environment, without any kind of optimization criteria.

With a geometric distribution, instead, the program tries to place the nodes in a geometric way in the environment, according to its the shape.

In the third case, with a signal-based distribution, the nodes are placed trying to optimize the signal spread over the environment. In this case, each insertion of a node in the area puts in doubt the previous placements if the signal could have been distributed in a better way.

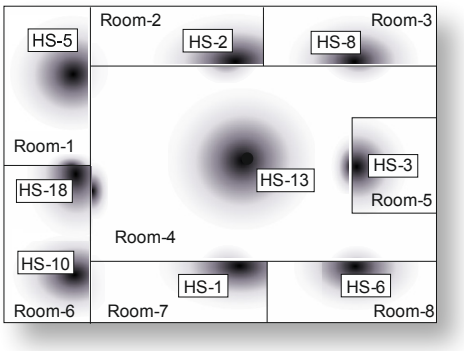

Figure 2: Example of a generated map with nodes and obstacles made by the simulation program

Subsequently, a vector for each point is generated, where each component represents the simulated signal power from a node of the network. By building the union of a specific component taken by all the points, we obtain a fingerprinting radio-map. In Figure 2, an example of how our simulation program generates the map is shown: we start from an environment divided into rooms surrounded by walls that surely alter the signal transmission. In order to have the best signal spread possible, hubs are placed trying to cover the greatest room of the environment till coverage of any room is guaranteed.

We show some code lines in order to explain how the simulation program works. It is executed in "mode 6", meaning some information are supposed to be given: parameters for map building, nodes position and obstacles features. The used variables are: *V* is the values variation percentage; *A* is the length of the map; *AN* is the number of points on a row; *B* is the height of the map; *BN* is the number of points on a column; *M* are the map points where the signal is evaluated; *OS1* and *OS2* are the central positions, radius and obstacles influence on the signal; *NOD* are nodes positions; *MPot* is the list of points signal powers from any node.

Procedure: Map generation

```
genmap(6,V,A,AN,B,BN,M,OS1,OS2,NOD,MF,MPot):-
    AN1 is A/AN,
    BN1 is B/BN,
    gengrid(M,A,AN1,B,BN1),
    map(Mpot,V,M,OS1,OS2,NOD),
    concatMeMpot(MF,M,MPot),!.
```

The procedure *gengrid* generates the list of points where we have to simulate the signal power; *map* creates the map with relative signal powers, taking one point at a time and generating for it a vector whose components depend on the nodes and obstacles positions; *concatMeMpot* concatenates the vectors of a list with those of another one. Now, the research algorithm which has to locate a point in the environment exploits the results of the simulation program. The behavior of our research algorithm can be summarized as shown in Figure 3.

Let us suppose that *P* is the point we want to find in the simulation of our algorithm: first, the procedure receives as input the signal power retrieved by the simulation program for the point *P* and the one for all the other points of the map. Then, the average (or alternatively the variance) is computed between the measurements and a comparison between the points of the map is performed, by using a tolerance τ given by the Mean Square Deviation (MSD). A list of coordinates of those points which satisfy the tolerance is obtained: if this list ends up being empty, the tolerance can be increased, by following the cyclic path; otherwise, the average between the coordinates of that list is computed in order to obtain a position for *P*, and the procedure terminates.

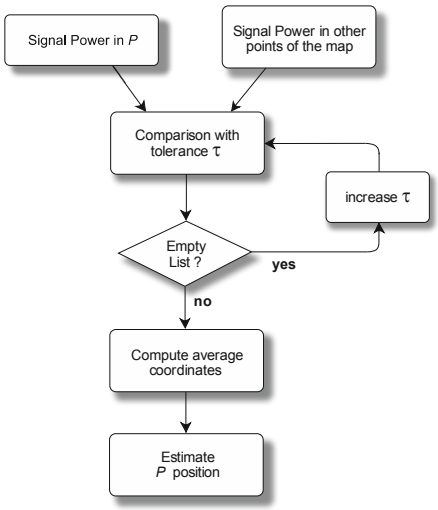

Figure 3: Control flow representing how the research algorithm works and finds an estimation of a point position

We show a code fragment below, where the research algorithm is explained in "mode 1", meaning that we assume that the map is given with points and power detections from any node retrieved in the point we want to find. First, we explain the used variables: *MF* represents map points with relative signal power from any node; *Pot* are measurements of power in the wanted point; *P* are coordinates close to the looked point; *List* is the list of map's points whose measurements are close to the wanted point; *Pots* is the tolerance; *Potm* is the average of powers of measurements in the point; *X* is the point we want to locate; *D* is the distance between the point we found and the one we wanted. Moreover, the procedure *average* computes the average between the vectors of powers; *meansqdev* computes the mean square deviation between vectors; *list* creates the list of points whose retrieved power is close to the one of the wanted point; *average1* computes the average of the list of points that should be close to the looked for point and it

increases the tolerance if needed; *dist* computes the distance between the found point and the wanted one.

Procedure: point research

```
search(1,..,MF,Pot,P,List,Pots,Potm,X,D):-
    average(Pot,Potm),
    meansqdev(Pot,Potm,Pots),
    list(List1,MF,Potm,Pots),
    average1(List1,List,MF,Potm,Pots,0,P),
    dist(X,P,D),!.
```

In this work we deal with localization problem by using a logic programming language: in both code pieces, we can see how the logic approach and Prolog programming language help us avoiding redundancy in computation. This is made thanks to the cut operator (!) that as soon as an advantageous computation branch is found, discards the other paths, in order to not analyze branches that would have led to useless solutions.

V. Evaluations and Case Studies

In this section, we provide some examples of how the model described above works and some analysis on it. First, we show in Figure 4 an example of how our simulation program fills a given environment by adding hub step by step.

A. Case Study

Let us suppose to have an environment with some obstacles as shown in Figure 4. Step by step, the program chooses where is preferable adding devices to the program have the best possible coverage. The program, as usually happens, places the first device in the middle of the hole environment, in order to have a good signal distribution. As we can observe, from Figures 4.1 and 4.b, the program tries to first cover the biggest area of the environment (the central one).

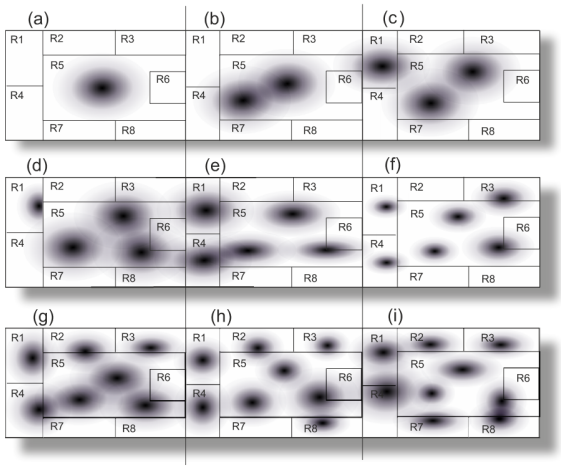

Figure 4: Evolution phases of the simulation program execution over an environment with some obstacles

In 4.c, the program decides to add a hub in one of the smaller rooms to have an improvement of signal distribution. After having covered the whole central area (Figure 4.d), the program starts adding devices in all the other rooms (Figures 4.e, 4.f, 4.g and 4.h), until it reaches the coverage of the entire environment and stops, as shown in Figure 4.i.

Moreover, we can see how the presence of obstacles determines a distortion in the signal shape, which is proportional to the attenuation factor α. This example is without any anchor nodes, but it is still possible adding some fixed node before the simulation start: in this case, clearly the addition of other hubs wouldn't have affected the position of anchors and thus it is likely that the final configuration of the network would have been different from the obtained one.

B. Evaluation

As we explained above, the second part of our work is the research algorithm, which estimates the position of a point in the environment, by exploiting the previous simulation program, based on fingerprinting radio-maps. We performed some analysis on how the variation of magnitude influences the precision of the algorithm. In particular, we consider three aspects: (*i*) number of involved nodes, (*ii*) grid dimensions, and (*iii*) environment size. In the following tables we show how the precision of our algorithm varies according to these parameters.

Table 1: variation of precision according to number of nodes

Nodes	**4**	**6**	**9**	**12**	**16**
Rand. Distr.	2,04m	1,72m	0,94m	0,59m	0,41m
Geom. Distr.	2,05m	1,45m	0,47m	0,33m	0,19m
Signal Distr.	2,02m	1,17m	0,48m	0,33m	0,2m

As we can observe from Table 1, the more the nodes are the more the precision of the algorithm grows. This does not mean that we can increase the number of nodes in an unchecked way, since we couldn't obtain an absolute precision: this is a consequence of the fact that measurements are made in map points which are in the detections grid too. This research gives as result all the points of the grid that are close to the point we are looking for.

Table 2: variation of precision according to grid dimensions

Row x Columns	**10x10**	**20x20**	**30x30**	**40x40**
Rand. Distr.	0,55m	0,41m	0,33m	0,29m
Geom. Distr.	0,42m	0,19m	0,15m	0,13m
Signal Distr.	0,43m	0,20m	0,15m	0,14m

As Table 2 shows, by increasing grid dimensions the precision increases. This is obvious, since there is a higher probability that points are close to the one we are looking for, during the comparison phase.

Table 3: variation of precision according to environment dimensions

Env. Dim. (mxm)	**5x5**	**10x10**	**15x15**	**20x20**
Rand. Distr.	0,12m	0,41m	1,16m	3,43m
Geom. Distr.	0,1m	0,19m	0,68m	1,84m
Signal Distr.	0,1m	0,2m	0,63m	1,97m

Finally, as we could expect, the more the environment grows the more the error increases, since each node influences just a small part of the entire environment, hence localization mistakes are more frequent (Table 3).

As we can clearly read by the tables above, the random distribution should be avoided, since it leads to less precise results; oppositely, both geometric and signal-based distributions provide solutions with a good precision, hence should be preferred to the random one.

VI. Conclusions and Future Work

In this work, we have initially described Wireless Sensor Networks and their main features and topology, with the aim to focus on the topic of point localization in indoor environments.

There are many techniques able to do this kind of localization, but in this work we try to propose a solution taking into account the environment features. Our localization algorithm is based on fingerprinting maps, and for this reason we have designed a simulation program which automatically generates them, rather than manually collect all useful information. These maps are generated by using RSS (Received Signal Strength) from any node, and we have considered three different node distributions: (*i*) random, (*ii*) geometric, and (*iii*) signal based. The approach is the logic one, based on backtracking: thanks to the Prolog cut operator (!), we are able to improve the computational effort of our algorithm, by pruning the useless computation paths, and just following the convenient ones. This technique allows us to provide the best solution according to the *global optimum*, rather than the *local* one: this is because at any insertion, previous decisions may possibly be changed, if a signal distribution improvement is supposed to be possible.

By analyzing our results, we have noticed that in order to have precise localization, a trade-off between environment dimensions, grid size, and number of involved nodes is necessary.

As hints for future works on this topic, we can observe that our simulation program can work in a 3D space as well, by providing adequate maps; moreover, it could also be opportunely integrated with a user interface and policy management tools [18], to facilitate its usage and control.

References

[1] Lewis, F. L. (2004). "Wireless sensor networks". Smart environments: technologies, protocols, and applications, 11, 46.

[2] Kumar, A., Shwe, H. Y., Wong, K. J., & Chong, P. H. (2017). Location-Based Routing Protocols for Wireless Sensor Networks: A Survey. *Wireless Sensor Network*, *9*(01), 25.

[3] Ssu, K. F., Ou, C. H., & Jiau, H. C. (2005). Localization with mobile anchor points in wireless sensor networks. *IEEE transactions on Vehicular Technology*, *54*(3), 1187-1197.

[4] Mao, G., Fidan, B., & Anderson, B. D. (2007). Wireless sensor network localization techniques. *Computer networks*, *51*(10), 2529-2553.

[5] Chintalapudi, K., Padmanabha Iyer, A., & Padmanabhan, V. N. (2010, September). Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (pp. 173-184). ACM.

[6] Parker, R., & Valaee, S. (2007). Vehicular node localization using received-signal-strength indicator. *IEEE Transactions on Vehicular Technology*, *56*(6), 3371-3380.

[7] Liu, C., Wu, K., & He, T. (2004, October). Sensor localization with ring overlapping based on comparison of received signal strength indicator. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on* (pp. 516-518). IEEE.

[8] Rong, P., & Sichitiu, M. L. (2006, September). Angle of arrival localization for wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on* (Vol. 1, pp. 374-382). IEEE.

[9] Severino, R., & Alves, M. (2007, June). Engineering a search and rescue application with a wireless sensor network-based localization mechanism. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a* (pp. 1-4). IEEE.

[10] Savvides, A., Park, H., & Srivastava, M. B. (2002, September). The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 112-121). ACM.

[11] Belloni, F., Ranki, V., Kainulainen, A., & Richter, A. (2009, March). Angle-based indoor positioning system for open indoor environments. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on* (pp. 261-265). IEEE.

[12] Priyantha, N. B., Balakrishnan, H., Demaine, E., & Teller, S. (2003, November). Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 340-341). ACM.

[13] Mourad, F., Snoussi, H., Abdallah, F., & Richard, C. (2009). Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, *57*(8), 3226-3239.

[14] Balzano, W., Del Sorbo, M. R., Murano, A., & Stranieri, S. (2016, November). A logic-based clustering approach for cooperative traffic control systems. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 737-746). Springer, Cham.

[15] Balzano, W., Del Sorbo, M. R., & Stranieri, S. (2016, March). A logic framework for c2c network management. In *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on* (pp. 52-57). IEEE.

[16] Balzano, W., Murano, A., & Stranieri, S. (2017). Logic-based clustering approach for management and improvement of VANETs. *Journal of High Speed Networks*, *23*(3), 225-236.

[17] Balzano, W., & Stranieri, S. (2017, November). LoDGP: A Framework for Support Traffic Information Systems Based on Logic Paradigm. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 700-708). Springer, Cham.

[18] Giordano, M., Polese, G., Scanniello, G., Tortora, G. (2010, February). A System for Visual Role-Based Policy Modelling: In *International Journal of Visual Languages and Computing*, Vol. 21, No. 1, Elsevier, pp. 41-64.

[19] Balzano, W., Murano, A., & Vitale, F. (2016, March). WiFACT--Wireless Fingerprinting Automated Continuous Training. In *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on* (pp. 75-80). IEEE.

[20] Balzano, W., Murano, A., & Vitale, F. (2016). V2V-en–vehicle-2-vehicle elastic network. *Procedia Computer Science*, *98*, 497-502.

[21] Balzano, W., Murano, A., & Vitale, F. (2018). SNOT-WiFi: Sensor network-optimized training for wireless fingerprinting. *Journal of High Speed Networks*, *24*(1), 79-87.