# A Logic User-Based Algorithm to Improve Node Distribution in Wireless Sensor Network

Balzano Walter
Dip. Ing. Elettrica e Tecnologie dell'Informazione
Università 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*—Localization in Wireless Sensor Networks (WSN) is a largely discussed research topic. Different solutions have been proposed to solve the localization problem over the time, exploiting techniques such as angle-based, range-based, and range-free. In our previous work, we proposed a logic range free algorithm able to uniformly distribute hubs around a given environment: this distribution is aimed to give the best coverage possible of the areas of interest. In this procedure, the presence of obstacles is taken into account, since it can impact the signal power: for this reason, an attenuation factor has been introduced to understand in which measure the obstacles modify the result. The algorithm is based on Prolog backtracking technique, which reflects the procedure of organizing the relative positions of nodes at each step, in order to optimize their distribution over the environment. The main goal of this work is to improve this approach, by considering not only the need to ensure a certain signal in each zone, but also to focus on the areas where a big usage from the clients is detected, in order to prevent the network saturation, without losing the coverage property. To this purpose, a displacement factor is introduced to vary the previous result in favor of a user-based distribution. The value of this factor has to reflect the optimum trade-off between coverage and user's usage need.**

***Keywords: WSN, localization, coverage, Prolog.***

## I. INTRODUCTION

A Wireless Sensor Network is made by nodes self-organized, broadcasting information and data all over the net. Due to the huge potential of such a network, 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. To find the best coverage possible for the area of interest, the localization technique used is an important aspect in research field.

The absence of obstacles makes the hub positioning in outdoor environment a not interesting problem, while in case of presence of walls, doors, or other obstacles, the positioning of nodes in a network can became a bottleneck in the localization field. Any kind of 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.

In our previous work [22]. We designed a range-free algorithm able to provide a map of the optimal hubs distribution over a given environment, in such a way that the best coverage possible is ensured. 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 arranged, and the position of each hub is computed again in order to optimize some metric on the signal. This approach provides a non-greedy algorithm whose solution is guaranteed to be the global optimum, rather than the local one. The provided algorithm presents two variants of execution, that differ in the choice of having *anchor* nodes or not.

Aware of the need to cover as much as possible the interest area, this approach does not consider the user's need: often, a given environment has not an equally distributed usage, but a more powerful signal can be needed in a room rather than another. For instance, if we consider an environment made by rooms, and one of these rooms is a pretty big warehouse, the first approach of our algorithm probably would place two hubs to cover that room but, by analysing the user's usage information turns out that the warehouse signal is never used: it's clear that using two hubs to cover a non-used area is a waste of resources, and in this work we want to avoid this situation. The aim is to consider the user's usage as an important parameter for the hub distribution, but not the only one: indeed, we do not want to lose the coverage property.

The rest of the work is organized as follows: Section II contains important aspects of literature that inspired our work, and other related research concerning localization issues in wireless sensor networks; in Section III, we explain the motivations that let us improve our previous work, by first recall all the main features of it, and then summarize the aspects that can be improved in our perspective; Section IV contains the very strategy to introduce the user's usage information in the approach, and how it is integrated with the coverage request in order to obtain the best possible solution that respects both the properties; finally, in Section V, the conclusions of this work and its advantages are shown.

## 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

DOI reference number 10.18293/VLSS2018-031

and structures monitoring, traffic management, surveillance, and many other 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 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 lies: 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 behaviour, 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.

The main inspiration for this work is given by our previous work [22], with the aim to improve it by considering an important metric not taken into account by now, which is the usage of the network that the client typically does.

First, we introduce Wireless Sensor Networks, a system of nodes which exchanges data wirelessly. All this information can be possibly held and elaborated by a control unit. As known, each net can have a particular topology, which characterizes the behaviour 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 neighbour.

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 unstable, 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.

## III. MOTIVATIONS

In our previous work [22], the localization problem is solved by means of a simulation program that has the aim to distribute nodes in a given environment according to a coverage criterion: we consider three possible 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.

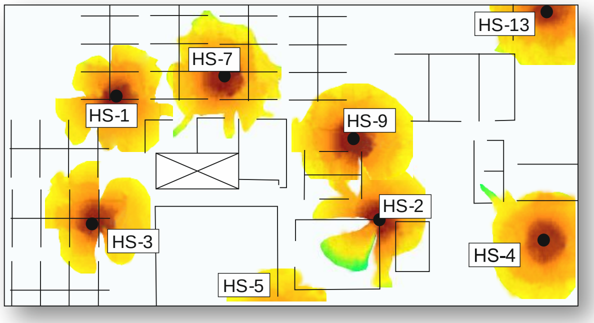

*Figure 1: Generic fingerprinting radio-map with different nodes and variation of signal due to presence of obstacles*

In any of these cases, the main goal is to obtain the best coverage possible in the environment, in such a way that it's not possible to find areas where the signal is not available (even if with a weak power). It 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 colour 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 colour 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).

For each point the power level is computed by using the inverse-square law:

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

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 $\alpha$ 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 the three different node distributions cited above.

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.

Step by step, the program chooses where is preferable adding devices in order to have the best coverage possible, as explained in [22]. The program, as usually happens, places the first device in the middle of the hole environment, in order to have a good signal distribution. At the very beginning, it tries to cover the biggest area of the environment, as we can observe from Figure 2.

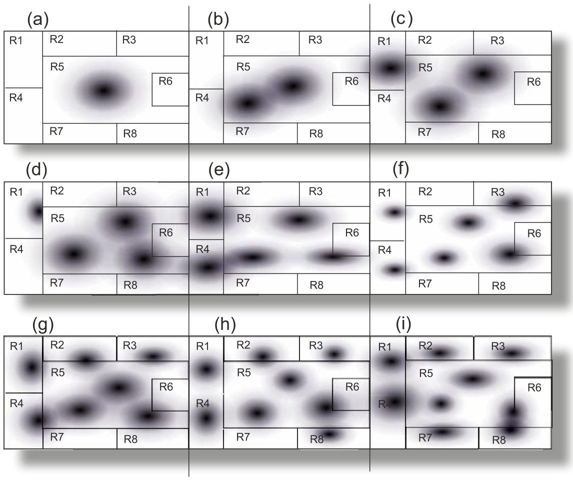

*Figure 2: Evolution phases of the simulation program execution over an environment with some obstacles*

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 $\alpha$. 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.

In some cases, a more meaningful criterion can be found to distribute nodes around a given environment, by considering the client needs: for instance, if we consider an environment made of some rooms, and one of them is actually a warehouse of big dimensions, but where very few signals of usage retrieved, our algorithm probably would place a certain number of hubs in that room in order to cover it, but this could be a redundancy if we take into account the user's usage. Indeed, some of the hubs used for the warehouse could have been better employed in areas where the signal is needed the more.

We analyzed the behavior of our algorithm, and the result highlights that its precision can vary according to number of nodes, grid dimension, and environment shape. In particular, 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.

Concerning the grid features, 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.

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.

Our final consideration is that 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.

Clearly, this approach can impact the coverage requirement: in some cases, the distribution of hubs according to the user's usage can decrease the environment coverage, by having some small areas not covered at all. But, the guess is that those not covered areas are surely not of interest for the users, and it is reasonable to reduce the coverage in not used areas in favour of those with a high density of users. In a border case, we could have whole rooms not covered at all, but this is not the goal of our improvement, since this would be a too heavy restriction. For this reason, we introduce a displacement factor $\delta$ that represents the percentage of how much the user's distribution influences the hub's positioning based on environment coverage. According to this factor, $\delta$=100% represents the border case explained above, where the distribution proposed by our starting algorithm, based on coverage, is ignored in favour of a distribution based exclusively on user's usage. The opposite case is given by $\delta$=0% that is essentially the same algorithm proposed in [22], where the only parameter, that is taken into account, is the environment coverage.

In a generic case, with $\delta$=50% for instance, the idea is that the first resulting distribution given by our simulation algorithm is slightly modified in order to improve the user's usage, by not losing the property of coverage guaranteed by the first approximation.

This improvement of our first approach [22] can seriously lead to a more efficient nodes distribution that provides not only the signal coverage in the whole environment, but also a stronger signal power where needed for the users, by finding the best value for the displacement factor.

In the next section, we are going to describe the mechanism that allows us to retrieve and exploit the user's usage statistics.

## IV. USER USAGE COMPENSATION

The information about how much a certain area is used by network's clients can be easily detected by using some utilities able to retrieve statistics concerning the usage during the time, such as *WirelessNetView* for the RSSI (Received Signal Strength Indicator) data detection.

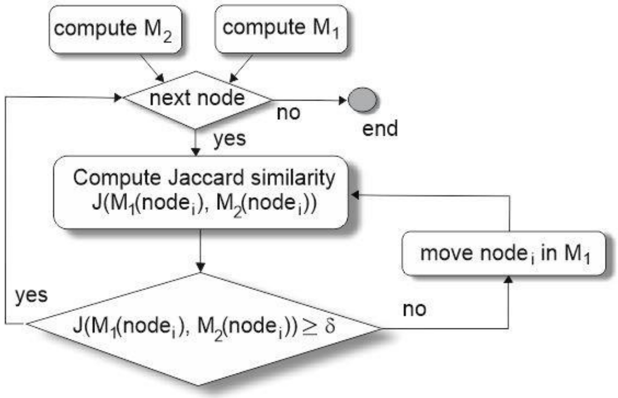

*Figure 3: flow of control with the user's usage statistics compensation with the aim to obtain a fixed minimum displacement with respect to the original solution based on coverage*

The idea is to exploit such information in order to build a map according exclusively to the user's usage of the network, similarly to how we build the map based on environment coverage. This is a border map, representing a displacement factor of 100%, which is not our aim, but it is useful in order to obtain the desired displacement value $\delta$. After the generation of the map based on coverage criterion, a second map has to be created, considering only the user's usage. These two maps represent the two opposite limits (δ=0% and δ=100%), but what we are going to compute is a third map that is the result of a trade-off between coverage and user's usage, with a displacement δ in the interval (0,100).

In order to do that, we indicate with $M_1$ the map based on coverage and with $M_2$ the one based on user's usage. We will exploit a largely used similarity measure, which is the Jaccard index. This indicator represents the similarity between the corresponding hubs in M1 and M2, by taking the intersection between the areas covered by the nodes and normalizing it with the union.

$$J(M_1(node_i), M_2(node_i)) = \frac{coverage(M_1(node_i))}{coverage(M_2(node_i))}$$

Where $M_1(node_i)$ represents the i-th hub in the first map, and $coverage(M_1(node_i))$ is the set of space points reached by the signal of the i-th node in the first map. Clearly, the Jaccard index is in the interval [0,1] (where 0 is a non-similarity indicator, and 1 is the maximum similarity possible). Clearly, we are not interested in obtaining a unitary similarity between the two maps, since this would possibly determine uncovered areas of the environment and this is not the best approach. What we want to obtain is a similarity based on the displacement value previously fixed, in particular:

$$J(M_1(node_i), M_2(node_i)) \geq \frac{\delta}{100}$$

Thus, the aim is to move a node from the first map $M_1$ in order to let it similar to the same node, but in the second map $M_2$, in such a way that their similarity is proportional to the displacement factor. Going through successive approximation of the map that keep moving the node toward the ideal position (according to the user's usage), the procedure stops when the minimum desired similarity is reached.

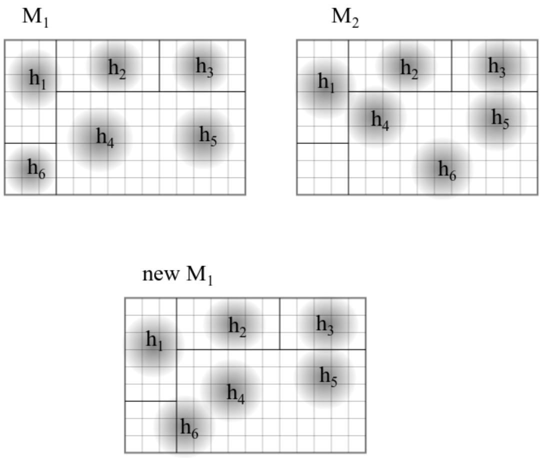

*Figure 4: example of how the two border maps are arranged to obtain a new map, which is a trade-off between coverage and usage, considering a displacement of 30%*

In Figure 3 is show the procedure explained by now: both the maps are generated: the first one only considers the coverage aspect, while the second one is based on the usage information of the network. The next steps are repeated until all the nodes of the network are considered. Each node is compared with the same node of the other map in order to compute the Jaccard similarity between them, according to the ratio between the points in the intersection of the nodes range and the ones in the union. Finally, the node is moved in the first map until the minimum desired displacement factor δ is obtained as Jaccard similarity between the two corresponding nodes that are being analysed. At the end of this procedure, the map $M_1$ will no longer reflect the nodes distribution aimed to maximize the environment coverage, but it will be the modified distribution that takes into account the usage of the network too, in a percentage induced by δ.

In Figure 4, an example of the possible result is shown: let's consider the square as unity of measure, and the box surrounding the hub (9 units in total) the points in the range of the corresponding hub. The map $M_1$ shows the hub's distribution according to the environment coverage, while the map $M_2$ the optimal distribution based on information detected concerning the usage of the network. In this example, we consider δ=30%, meaning that for each pair of corresponding hubs we need a Jaccard similarity of at least 0.3. Intuitively, the hubs $h_1$, $h_2$ and $h_3$ respects this similarity, thus we focus on the remaining three hubs $h_4$, $h_5$ and $h_6$.

Looking at the hub $h_4$ in $M_1$ and $M_2$, we can see that their intersection is 4 (the number of squares that they share), while the union is 18 (twice the number of squares in the range, that is 9 as we said): this means that the Jaccard similarity between them is 4/18=0.2, which is not enough with respect to δ. Indeed, in the new map $M_1$ the hub $h_4$ is moved in such a way that the similarity reaches the desired one.

The hubs $h_5$ in $M_1$ and $M_2$ have an intersection of 6 points, and the union is always 18, as before. Thus, the Jaccard similarity is 6/18=0.3, which is acceptable, and this is the reason why it is not moved from the starting position in $M_1$.

Finally, hubs $h_6$ no intersection, indeed the movement is more evident with respect to $h_4$ since it was not similar at all with its corresponding in $M_2$.

The final configuration is the one shown in new $M_1$, where all the similarities between corresponding hubs is at least 0.3.

**Procedure:** usage compensation

```
$map_1$(node,position).
$map_2$(node,position).

check_jaccard(node,δ):-
    J($map_1$(node,$p_1$),$map_2$(node,$p_2$)>=δ/100,!.

check_jaccard(node,δ):-
    J($map_1$(node,$p_1$),$map_2$(node,$p_2$))<δ/100,
```

```
    move(node,map1),check_jaccard(node,δ).
```

The procedure shown in the code fragment above shows exactly the way this approach works: let's suppose of having a Prolog fact for each node of the maps that says the position of that node in the corresponding map.

The predicate *chech_jaccard(node,δ)* checks if the Jaccard similarity is at least the minimum desired one (δ/100), otherwise the corresponding node is moved in the map $M_1$ and the Jaccard similarity is computed again, until the constraint is satisfied.

In this work we deal with localization problem by using a logic programming language: 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 analyse branches that would have led to useless solutions.

It is clear that without any further check, this successive approximation of nodes position inside the map $M_1$ can progressively let a room to be free of any kind of signal: this is what we wanted to avoid in our starting considerations. To avoid this kind of behaviour, we have to check that the next node movement does not impact the coverage of a whole room, before it is performed. If this happens, we have two possibilities:

- Reduce the displacement factor δ in order to relax the minimum required similarity;
- Increase the resources: add a new hub in the room that becomes free of signal.

If the resources cannot be increased, there is no other possibility but decrease the displacement factor so that the next movement (that would cause the loss of signal in some room) doesn't have to be performed at all.

The Prolog code needs to be modified as follows:

**Procedure:** usage compensation with additional check

```
map1(node,position).
map2(node,position).

check_jaccard(node,δ):-
    J(map1(node,p1),map2(node,p2)>=δ/100,!.

check_jaccard(node,δ):-
    J(map1(node,p1),map2(node,p2))<δ/100,
    try_movement(m,node,map1),empty_areas(m),
    decreaseδ_increaseResources(map1),!.

: check_jaccard(node,δ):-
    J(map1(node,p1),map2(node,p2))<δ/100,
    try_movement(m,node,map1),!empty_areas(m),
    move(node,map1),check_jaccard(node,δ).
```

As shown in the code, an additional check needs to be performed: the movement of the node in the map is tried before being performed. If projecting the current movement some areas end up being uncovered, the movement is not performed and the procedure to decrease the displacement factor or increase the number of available hubs is executed. Otherwise, the movement is effectively performed, and the procedure is recursively called.

## V. CONCLUSIONS

One of the most important issues when one deals with wireless sensor networks is localization: not only it is crucial to locate objects all over the network, but it is essential to understand where hub should be placed in order to guarantee that some metrics are respected and exploited. Clearly, many proposals have been made in literature concerning the localization topic but, in this and our previous work, we use a logic approach based on Prolog facts and rules to simulate the hubs positioning over the interesting environment, based on successive approximation of nodes distribution: the relative position of each node is opportunely arranged at each iteration in such a way that the metric taken into account is respected.

While in our previous work we focus on the coverage property, meaning that the hubs distribution was only aimed to optimize the signal spreading in each area of the reference environment, in this work we improve this approach by adding another meaningful metric: the usage of the network.

Intuitively, the resources supply can be better performed if we focus on the areas where the usage of the network is very high, rather than areas where no one uses the provided signal.

From the other hand, a displacement in favour of usage statistics can seriously impact the coverage property, meaning that with this approach whole rooms could be not covered at all by any kind of signal. This is obviously something to avoid: for this reason, we introduce a displacement factor, which represents how much one is ready to sacrifice the coverage in favor of a better user-based distribution of the signal. We have shown some Prolog code lines that explain how this strategy works: two maps are used, the first one is the one resulting from our previous approach, and the second one is the one with a hundred percent of displacement factor. None of them constitute the final solution, which is rather a trade-off between them: indeed, by using the Jaccard similarity measure we managed to have a final map with a minimum displacement factor required, but without losing the coverage metric.

We went through two approximations of our solution: first, we have considered the possibility to move the nodes in the first map according to the displacement factor which determines the minimum similarity required for each pair of corresponding nodes in the two maps. This solution clearly solves the problem of introducing the user's information in the node distribution, but still does not guarantee that the coverage property is kept: this is the reason why the second approximation has been done, by introducing an additional check, ensuring that no areas can became free of signal at all.

## 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.

[22] Balzano, W. Stranieri, S. (2018). A Logic Range-Free Algorithm for Localization in Wireless Sensor Networks, in *The 24th International DMS Conference on Visualization and Visual Languages*, San Francisco, California, 2018. DOI reference number: 10.18293/DMSVIVA2018-008.