A Secure and Low Cost Range-Free Localization Algorithm for Mobile Sensor Networks

Anju Thomas
M. Tech. Student
Department of Computer Science & Engineering
Nehru College of Engineering and Research Centre, Pampady, Thrissur

Remya Ramachandran
Assistant Professor
Department of Computer Science & Engineering
Nehru College of Engineering and Research Centre, Pampady, Thrissur

Abstract

In these days, it is very important to trace a mobile in the vast mobile sensor network. It is mainly for identifying the location of the person who is using that mobile. For this application, the location of the mobile node (mobile) is to be found. This process is known as localization. Out of different types of localization techniques, range-free localization approaches are cost-effective for mobile sensor networks (because no additional hardware support is required). Due to economic considerations, mobile sensor networks typically have sparse anchor nodes which make most range-free localization algorithms inaccurate. On the other hand, due to the power limitation of mobile sensor nodes (i.e., they are battery-operated) and high power consumption by communication, high communication cost will significantly reduce the network life time. For solving these two problems, historical beacons (i.e., anchor nodes' announcements delivered in previous time slots) and received signal strength (RSS) are used to derive three constraints. By the aid of the three constraints, introduced a low-communication-cost range-free localization algorithm (only one-hop beacon broadcasting is required).Also extended this project by adding a data discovery and dissemination protocol. This makes the system free from vulnerabilities.

Keywords: Localization, Anchor Nodes, Beacon, Dissemination, Cryptography

I. INTRODUCTION

Localization is a critical issue in wireless sensor networks (WSNs). Although GPS has been widely used to assist location-based services, it is impractical to equip each sensor node with a GPS device in large-scale WSNs. Therefore, localization algorithms for WSNs typically use a limited number of anchor nodes, which are aware of their locations, e.g., by the aid of GPS, while the other nodes (referred to as normal nodes) estimate their locations using the location information of anchor nodes. Such localization algorithms are anchor node-based, and they can be further divided into two categories: range-based and range-free.

A range-based localization algorithm calculates locations with absolute point-to-point distances, while a range-free localization algorithm calculates locations without these distances. However, distance estimation techniques usually require additional expensive hardware support (e.g., angle of arrival (AoA) and time difference of arrival (TDoA)), or have low accuracy (e.g., received signal strength (RSS)-based approaches). Due to the hardware limitations of WSNs, range-free solutions are being pursued as an alternative to range-based solutions. Most of prior range-free localization algorithms were designed for static sensor networks and not applicable to mobile ones. Existing range-free localization approaches for mobile sensor networks usually suffer from sparse anchor node problem and high communication cost. Due to economic considerations, wireless sensor networks typically have sparse anchor nodes which makes most range-free localization algorithms inaccurate. On the other hand, in mobile sensor networks, sensor nodes are battery-operated and communication is the highest power consumption item. Prior localization algorithms achieve the required accuracy with high communication cost and high communication cost will significantly reduce the network life time. Moreover, due to the rapid development of wireless technologies (e.g., Wi-Fi and Bluetooth) and quickly emerging applications, the ISM band, which is used by most WSNs, has become crowded and congested. Hence, localization algorithms with high communication cost will be impractical in the near future.

II. RELATED WORK

Today, smart environments are deployed everywhere, and mobile sensor networks can be used in many different scenarios. Mobile sensor networks are particularly interesting in hazardous or remote environments, or when a large number of sensor nodes have to be deployed. The localization issue is important where there is an uncertainty about some positioning. An effective localization algorithm can then use all the available information from the motes to compute all the positions. The aim of this paper is to develop an algorithm for localization of nodes in a sensor network. The algorithm should be distributed and executed in individual nodes; schemes that pool all data from the network and perform a centralized computation will not be considered. Since the algorithm should be run in individual sensor nodes, the solution has to be relatively simple, and demand limited resources (in terms of computation, memory and communication overhead).

A. Range Based Methods

The range methods exploit information about the distance to neighbouring nodes. Although the distances cannot be measured directly they can, at least theoretically, be derived from measures of the time-of-flight for a packet between nodes, or from the signal attenuation. The simplest range method is to require knowledge about the distances to three nodes with known positions (called anchors or beacons depending on the literature), and then use triangulation. However, more advanced methods exist, that require less severe assumptions.

B. Range-Free Methods

Regarding localization, it uses fusion of RF received signal strength information and acoustic time of flight. There is an interesting definition of a distributed algorithm for random WSN. The minimal density of known nodes is presented. The main objective of their algorithm is to broadcast a request (*"Do you hear me?"*) and compute the estimated localization by the interpretation of the answer of all the known nodes. The influence of noise can be important, shows (flip and flex ambiguities). To minimize it, Robust Quadrilaterals and Clusters are defined and analysed. But the computation complexity increases as it is extended to large-scale WSN, which is a big inconvenient. Localization schemes that exploit the additional information that can be obtained when some nodes are mobile. Three schemes are possible: static nodes - moving seeds, moving nodes - static seeds, or both moving.

By knowing the original emitted power and comparing it to the received signal power, one can estimate the attenuation g and deduce the distance via, for example, a free space path-loss model:

$$g = d^{\wedge}\alpha$$

In this scheme the exponent α is around 2 in an open-space environment, but its value increases if the environment is more complex (walls, etc.) or less suitable for radio waves (metallic devices...). Another issue is that there is no unique path from the transmitter to the receiver. Any reverberations of the signal will influence the received strength, so it has to be measured at the appropriate moment. Some consider the first peak, whereas others prefer an average of the first periods. Contrary to the first ones, those methods never compute the distances to the neighbours. They use hearing and connectivity information to identify the nodes and beacons in their radio range, and then estimate their position. Do you hear me ? This idea of only using the information of the immediate neighbours fits perfectly the distributed approach of the localization problem. In those type of schemes, every node only uses direct communications to refine his position estimates, and when it succeeds to achieve a given accuracy, it broadcasts the result. The big advantage is that it saves a lot of traffic, but an overload of the radio channels can occur. This has to be carefully studied, and the rules for priority clearly established. Another drawback is the fact that those techniques usually require a great amount of nodes.

III. THE SYSTEM MODEL

This work starts by analysing a range-free localization algorithm, HitBall, for mobile sensor networks. In this algorithm, mobile sensor networks are assumed to have mobile normal nodes and mobile anchor nodes. Each anchor node is assumed to broadcast a beacon that carries its location information to its one-hop neighbouring normal nodes per slot (i.e., one-hop-beacon broadcasting). Each normal node a can determine the possible region (of its location) by the aid of collected beacons. A possible region of a's location is a region which covers a's location. Clearly, a smaller possible region implies higher localization accuracy. A beacon is called a current beacon if it is delivered in the current time slot, and a historical beacon otherwise (i.e., prior to the current time slot). Associated with each current beacon, there is a one-hop-anchor-constrained region, which is the communication range of the anchor node when it sent out the beacon. Besides, associated with each historical beacon, there is a historical-anchor-constrained region, which is a circle centered at the anchor node that sent out the beacon. If the historical beacon was delivered t time slots ago, then the circle has a radius of $r + V_{max} * t$, where r is the communication radius of an anchor node and V_{max} is the maximum moving distance of a normal node during a time slot.

The HitBall algorithm and MCL-based range-free localization algorithms determine the possible region of a's location by finding the intersection of all one-hop-anchor-constrained regions of a with others constrained regions (e.g., historical-anchor constrained regions and ring areas centered at two-hop neighboring anchor nodes of a). In these algorithms, more constrained regions can determine a smaller possible region of a's location and hence improve the localization accuracy. In this paper, we introduce three RSS-constrained regions for a. Our possible region of a's location is the intersection of one-hop-anchor-constrained regions, historical-anchor-constrained regions, and the proposed three RSS-constrained regions.

Let $b_{t,i}$ denote a beacon which is received by normal node a in time slot t from an anchor node b_i. If a receives beacons $b_{1,1}$, $b_{1,2}$, $b_{1,3}$, and $b_{2,3}$ in the first two time slots, then there are two historical beacons (i.e., $b_{1,1}$ and $b_{1,2}$) with respect to time slot 2 and four historical beacons with respect to time slot 3. For a node p, let $L_t(p)$ be the location of node p in time slot t. For a beacon $b_{t,i}$,let $L_t(b_{t,i})$ be the location of the anchor node b_i when it sent out $b_{t,i}$ and let $L_t(b_{t,i})$=null if $j \neq t$. This algorithm is a refinement of MCL and consists of three phases: sample generating phase, sample filtering phase, and location estimation phase. In the sample generating phase, a determines samples from the intersection of all one-hop-anchor-constrained regions and historical-anchor constrained regions. However, the intersection mentioned above is difficult to calculate for resource-limited normal nodes. Thus, in our algorithm, each one-hop-anchor constrained region Xc (or historical-anchor-constrained region Xh) is replaced with a minimum square Sc (or Sh) which can cover Xc (or Xh). In the sample filtering phase, normal node a filters out invalid samples

by the aid of the proposed RSS constrained regions. In the location estimation phase, a estimates $L_k(a)$ to be the centroid of all valid samples.

Algorithm 1: The HitBall Algorithm

/* Suppose that a wants to determine its location in slot k */

Sample Generating

1) For each current beacon $b_{k,i}$ (i.e., received from anchor b_i in slot k), determine the minimum box, called Sc, which can cover the one-hop-anchor-constrained region associated with $b_{k,i}$.
2) For each historical beacon $b_{j,i'}$ (i.e., received from anchor b'_i in slot j), determine the minimum box, called Sh, which can cover the historical-anchor-constrained region associated with $b_{j,i'}$.
3) Let rectangle I be the intersection of all Scs and Shs.
4) Rectangle I is divided into m squares, where m is the number of needed samples. The central point in each square is chosen as a sample.

Sample Filtering

5) Construct an RSS-constrained region for each beacon pair.
6) A sample is valid if it is inside all RSS-constrained regions constructed in step 5. Location Estimation
7) Estimate Lk(a) to be the centroid of all valid samples.

After a wireless sensor network (WSN) is deployed, there is usually a need to update buggy/old small programs or parameters stored in the sensor nodes. This can be achieved by the so-called data discovery and dissemination protocol, which facilitates a source to inject small programs, commands, queries and configuration parameters to sensor nodes. It is different from the code dissemination protocols (also referred to as data dissemination or reprogramming protocols), which distribute large binaries to reprogram the whole network of sensors. For example, efficiently disseminating a binary file of tens of kilobytes requires a code dissemination protocol while disseminating several two-byte configuration parameters requires data discovery and dissemination protocol. Considering the sensor nodes could be distributed in a harsh environment, remotely disseminating such small data to the sensor nodes through the wireless channel is a more preferred and practical approach than manual intervention. More importantly, all existing data discovery and dissemination protocols employ the centralized approach in which, as shown in the top sub-figure in Figure, data items can only be disseminated by the base station. Unfortunately, this approach suffers from the single point of failure as dissemination is impossible when the base station is not functioning or when the connection between the base station and a node is broken. In addition, the centralized approach is inefficient, non-scalable, and vulnerable to security attacks that can be launched anywhere along the communication path. Even worse, some WSNs do not have any base station at all. For example, for a WSN monitoring human trafficking in a country's border or a WSN deployed in a remote area to monitor illicit crop cultivation, a base station becomes an attractive target to be attacked. For such networks, data dissemination is better to be carried out by authorized network users in a distributed manner.

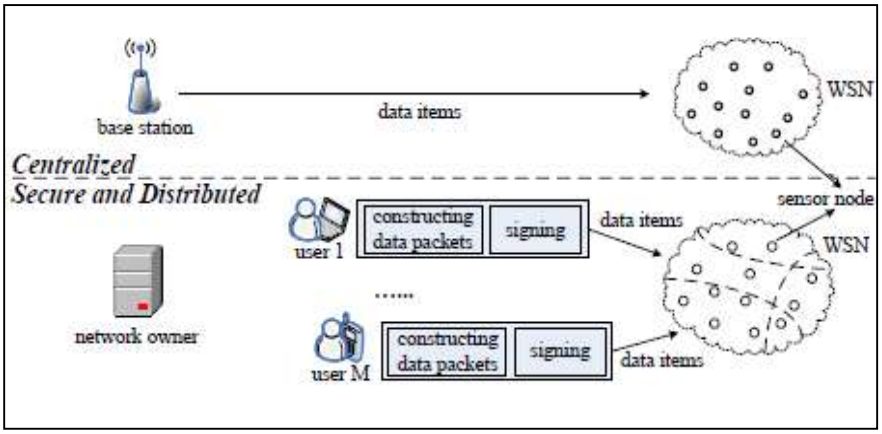


Fig. 1: system overview of centralized and distributed data discovery and dissemination approaches

Motivated by the above observations, this work has the following main contributions:

- The need of distributed data discovery and dissemination protocols is not completely new, but previous work did not address this need. We study the functional requirements of such protocols, and set their design objectives. Also, we identify the security vulnerabilities in existing data discovery and dissemination protocols.
- Based on the design objectives, we propose DiDrip. It is the first distributed data discovery and dissemination protocol, which allows network owners and authorized users to disseminate data items into WSNs without relying on the base station. Moreover, our extensive analysis demonstrates that DiDrip satisfies the security requirements of the protocols of its kind. In particular, applied the provable security technique to formally prove the authenticity and integrity of the disseminated data items in DiDrip.
- Demonstrated the efficiency of Di Drip in practice by implementing it in an experimental WSN with resource limited sensor nodes. This is also the efficient implementation of a secure and distributed data discovery and dissemination protocol.

A. Design Consideration of Secure and Distributed Data Discovery & Dissemination

As shown in the bottom subfigure in Figure.1, a general WSN comprises a large number of sensor nodes. It is administrated by the owner and accessible by many users. The sensor nodes are usually resource constrained with respect to memory space, computation capability, bandwidth, and power supply. Thus, a sensor node can only perform a limited number of public key cryptographic operations during the lifetime of its battery. The network users use some mobile devices to disseminate data items into the network. The network owner is responsible for generating keying materials. It can be off-line and is assumed to be uncompromisable. Networks users are assigned dissemination privileges by the trusted authority in a PKI on behalf of the network owner. However, the network owner may, for various reasons, impersonate network users to disseminate data items.

B. DIDRIP

Referring to the lower sub-figure in Fig. 1, DiDrip consists of four phases, system initialization, user joining, packet pre-processing and packet verification. For our basic protocol, in system initialization phase, the network owner creates its public and private keys, and then loads the public parameters on each node before the network deployment. In user joining phase, a user gets the dissemination privilege through registering to the network owner. In packet pre-processing phase, if a user enters to the network and wants to dissemination some data items, he/she will need to construct the data dissemination packets and then send them to the nodes. In packet verification phase, a node verifies each received packet. If the result is positive, it updates the data according to the received packet.

1) System Initialization Phase

In this phase, an ECC is set up. The network owner carries out the following steps to derive a private key x and some public parameters fy; Q; p; q; h(:)g. It selects an elliptic curve E over GF(p), where p is a big prime number. Here Q denotes the base point of E while q is also a big prime number and represents the order of Q. It then selects the private key x2GF(q) and computes the public key y = xQ. After that, the public parameters are preloaded in each node of the network. We consider 160-bit ECC as an example. In this case, y and Q are both 320 bits long while p and q are 160 bits long.

2) User Joining Phase

This phase is invoked when a user with the identity UIDj , say Uj , hopes to obtain data discovery and dissemination privilege. User Uj chooses the private key SKj2GF(q) and computes the public key PKj = SKj _Q. Here the length of UIDj is set to 2 bytes, in this case, it can support 65,536 users. Similarly, assume that 160-bit ECC is used, PKj and SKj are 320 bits and 160 bits long, respectively. Then user Uj send a 3-tuple <UIDj ; Prij ; PKj> to the network owner, where Prij denotes the dissemination privilege of user Uj . Upon receiving this message, the network owner generates the certificate Certj . A form of a certificate consists of the following contents: Certj = fUIDj ; PKj ; Prij ; SIGxfh(UIDjkPKjkPrij)g, where the length of Prij is set to 6 bytes, thus the length of Certj is 88 bytes.

3) Packet Pre-Processing Phase

Assume that a user, say Uj , enters the WSN and wants to disseminate n data items: di = fkeyi; versioni; dataig, i = 1; 2; : : :; n. For the construction of the packets of the respective data, we have two methods, i.e., data hash chain and the Merkle hash tree [16]. For data hash chain approach, a packet, say Pi is composed of packet header, di, and the hash value of packet Pi+1 (i.e., Hi+1 = h(Pi+1)) which is used to verify the next packet, where i = 1; : : :; n □ 1. Here each cryptographic hash Hi is calculated over the full packet Pi, not just the data portion di, thereby establishing a chain of hashes. After that, user Uj uses his/her private key SKj to run an ECDSA sign operation to sign the hash value of the first data packet h(P1) and then creates an advertisement packet P0, which consists of packet header, user certificate Certj , h(P1) and the signature SIGSKj fh(P1)g. Similarly, the network owner assigns a pre-defined key to identify this advertisement packet. With the method of Merkle hash tree, user Uj builds a Merkle hash tree from the n data items in the following way. All the data items are treated as the leaves of the tree. A new set of internal nodes at the upper level is formed; each internal node is computed as the hash value of the concatenation of two child nodes. This process is continued until the root node Hroot is formed, resulting in a Merkle hash tree with depth D = log2(n). Before disseminating the n data items, user Uj signs the root node with his/her private key SKj and then transmits the advertisement packet P0 comprising user certificate Certj , Hroot and SIGSKj fHrootg. Subsequently, user Uj disseminates each data item along with the appropriate internal nodes for verification purpose. Note that as described above, user certificate Certj contains user identity information UIDj and dissemination privilege Prij . Before the network deployment, the network owner assigns a predefined key to identify this advertisement packet.

4) Packet Verification Phase

When a sensor node , say Sj , receives a packet either from an authorized user or from its one-hop neighbours, it first checks the packet's key field (1) If this is an advertisement packet (P0=fCertj ; h(P1); SIGSKj fh(P1)gg for the data hash chain method while P0=fCertj ; root; SIGSKj frootgg for the Merkle hash tree method), node Sj first pays attention to the legality of the dissemination privilege Prij . For example, node Sj needs to check whether the identity of itself is included in the node identity set of Prij . If the result is positive, node Sj uses the public key y of the network owner to run an ECDSA verify operation to authenticate the certificate. If the certificate Certj is valid, node Sj authenticates the signature. If yes, for the data hash chain method (respectively, the Merkle hash tree method), node Sj stores <UIDj ;H1> (respectively, < UIDj ; root >) included in the advertisement packet; otherwise, node Sj simply discards the packet. (2) Otherwise, it is a data packet Pi, where i = 1; 2; : : : ; n). Node Sj executes the following procedure. For the data hash chain method, node Sj checks the authenticity and integrity of Pi by

comparing the hash value of Pi with Hi which has been received in the same round and verified. If the result is positive and the version number is new, node Sj then updates the data identified by the key stored in Pi and replaces its stored <round, Hi> by <round, Hi+1>(Hi+1 is included in packet Pi); otherwise, Pi is discarded. For Merkle hash tree method, node Sj checks the authenticity and integrity of Pi through the already verified root node received in the same round. If the result is positive and the version number is new, node Sj then updates the data identified by the key stored in Pi; otherwise, Pi is discarded.

C. *Block Diagram*

The figure 2 is the detailed block diagram for localization of mobile sensor networks. This localization algorithm has mainly three steps:

1) Sample generating
2) Sample filtering
3) Location estimation

In the first step, ie, sample generating a vast number of samples are generated. In second step, ie, sample filtering , a certain number of samples are filtered out which is not our desired location points for sure. In the third step, the desired location is estimated.

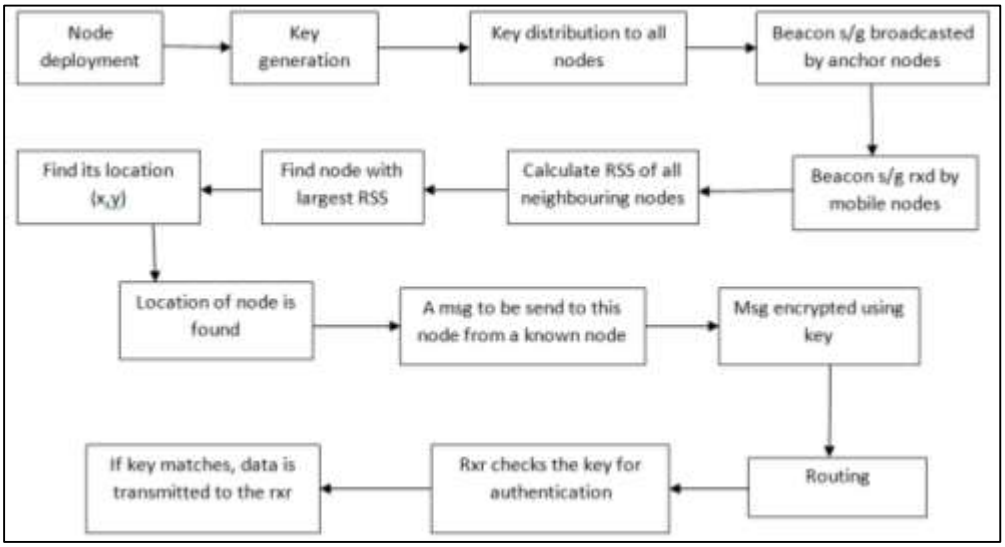


Fig. 2: block diagram of localization algorithm

In practical scenarios, the number of nodes is very much large. So the nodes are further classified into anchor nodes and mobile nodes. And also there are some sensor nodes. Anchor nodes is the representation of the base stations itself. Moving nodes are the mobile phones or other devices used by us. Sensor nodes are placed to receive beacon signals. The first step is the deployment of different types of nodes in the network. Anchor nodes are placed in a certain distances. Sensor nodes are placed in such a way that it gives boundary to anchor nodes. The next step is key generation. By using Diffie Hellman key exchange a secret key is generated in all the nodes in the network. Anchor nodes continuously sends signals to the environment. This signals are received by moving nodes. And sends back to the sensor nodes. By analysing the RSS (received signal strength) of all the nodes, the sensor node most nearer to the required moving node can be found. That is, the node with the largest received signal ratio. Now the location of the node is found. The next is to send a message to another node. For this , routing is used. In this work , DSR routing is used. The message is sent and received only after the verification of the key, so that ensure that the message is not sent to a unintended person. A data discovery and dissemination protocol is added to this work, to update some data or program in the nodes. For this purpose, RSA encryption decryption algorithm is used.

D. *Software Description*

NS2 is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy in this document), and a similar class hierarchy within the OTcl interpreter (also called the interpreted hierarchy in this document). The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. The root of this hierarchy is the class TclObject. Users create new simulator objects through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. The interpreted class hierarchy is automatically established through methods defined in the class TclClass. user instantiated objects are mirrored through methods defined in the class TclObject. There are other hierarchies in the C++ code and OTcl scripts; these other hierarchies are not mirrored in the manner of TclObject.

IV. EXPERIMENTAL RESULTS

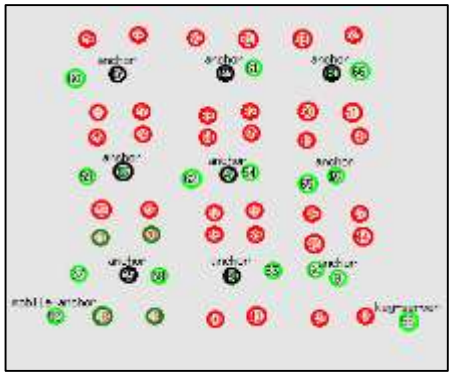


Fig. 3: Deployment of mobile sensor nodes

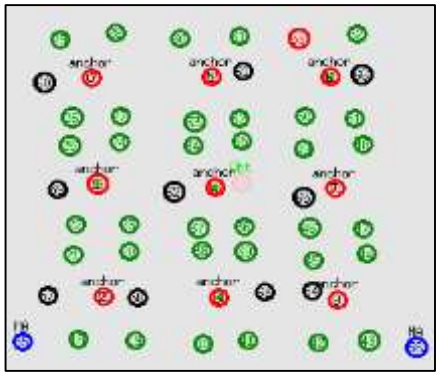


Fig. 4: Key distribution

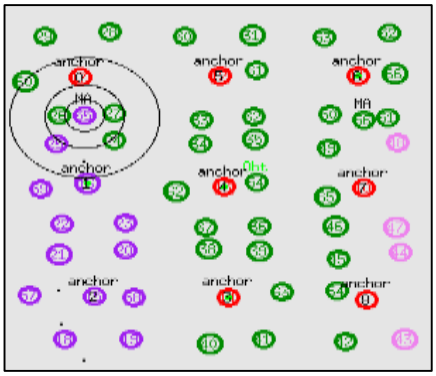


Fig. 5: Mobility of nodes in the network

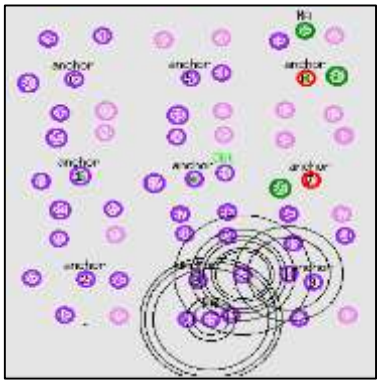


Fig. 6: broadcasting beacon signals and RSS measurement

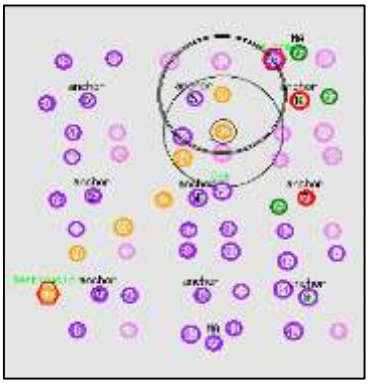
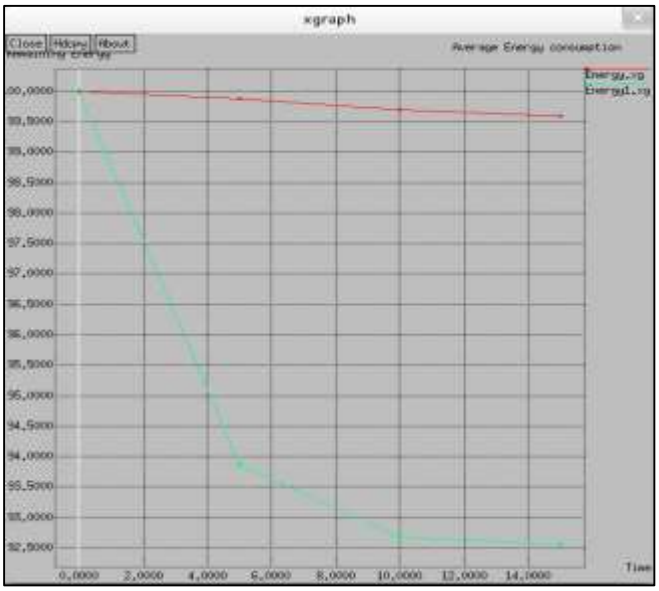
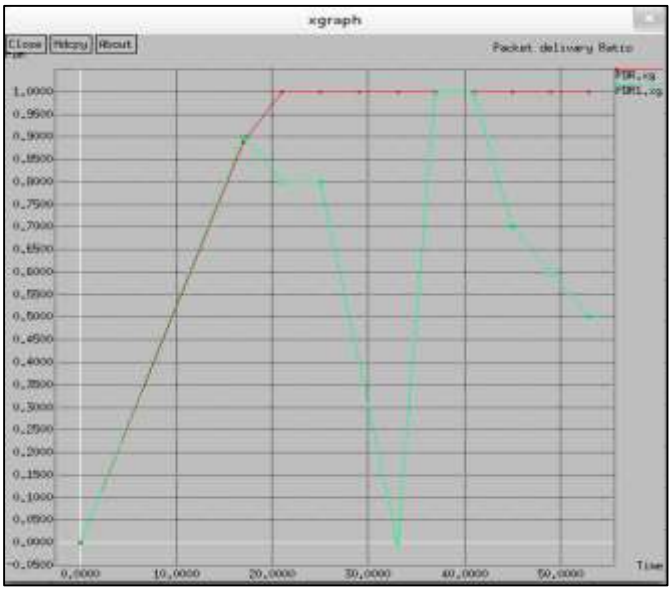


Fig. 7: Routing

E. Graphical Results





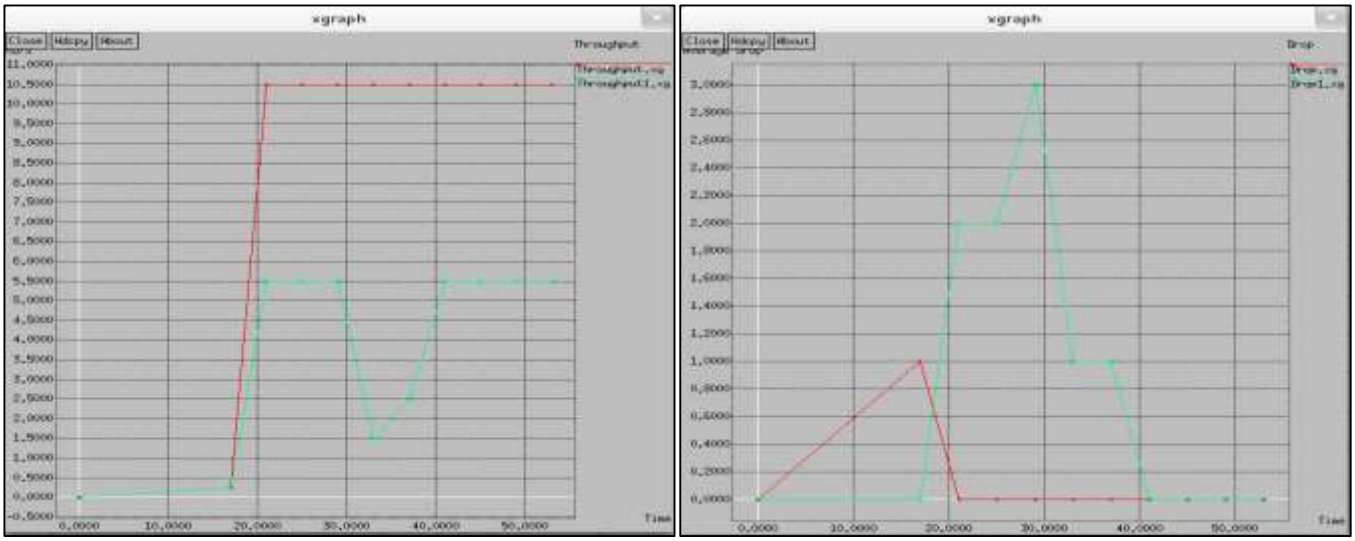


Fig. 8: Graphical Result

V. CONCLUSION

As wireless sensor networks became a key technology and are used in more and more industrial and environmental problems, defining an effective localization algorithm became an important task. The literature survey work showed that an optimum algorithm could not be defined yet, and thus the choice of the suitable one has to be founded on the specificities of the situations, taking into account the size of the network, as well as the deployment methods and the expected results.

Continuing further with the ideas, the algorithm studied the influence of different parameters in terms of performances. This lead to the major developments that have been proposed in the thesis, and which improvements significantly increase the positioning accuracy. They do not require much more computational costs, and perfectly match the distributed algorithm's requirements. A third aspect of the work concerned the feasibility of approaches based on received signal strength indication. Indeed, this showed that the values are really sensitive to the environment and the nature of the disturbances. The data recovery and dissemination protocol is added with the localization process and as a result the results of the work got improved. Hence got an efficient protocol.

Due to the important amount of scenarios where localization processes comes into play, it has been and will continue being an important research field. The different types of algorithms might be improved, and new ones developed, but another issue will be to determine which algorithm could give the more interesting results, for a given configuration.

REFERENCES

[1] L. Blazevic, J.-Y. Le Boudec, and S. Giordano, "A location-based routing method for mobile ad hoc networks," IEEE Trans. Mobile Comput., vol. 4, no. 2, pp. 97–110, Mar./Apr. 2005.
[2] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: A survey," IEEE Pervasive Comput., vol. 9, no. 1, pp. 38–47, Jan.–Mar. 2010.
[3] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," IEEE Trans. Comput., vol. 53, no. 3, pp. 241–250, Mar.2004.
[4] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," IEEE J. Select. Areas Commun., vol. 24, no. 2, pp. 247–260, Feb. 2006.
[5] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in Proc. IEEE Annu. Int. Conf. Comput. Commun., 2003, vol. 2, pp. 1293–1303.