Novel Localization Framework Algorithm for Wireless Sensor Network: Theoretical Analysis and Distributed Implementation using Contiki OS

Hussein Olayan*, Sofiène Affes*†, and Nahi Kandil†
*INRS-EMT, Université du Québec, Montreal, QC, H5A 1K6, Canada, Email: {elassaf,hussein.olayan, affes}@emt.inrs.ca
†LRTCS, University of Quebec in Abitibi-Témiscaming, Rouyn-Noranda, QC, J9X 5E4, Canada, Email: nahi.kandil@uqat.ca

***Abstract*—Localization algorithm has been critical and challenging topic for wireless sensor networks(WSNs). In this paper, an accurate localization algorithm tailored for WSN to compute the location of WSN is proposed. Using the proposed algorithm, the regular node (i.e unknown positions) will directly estimate its distance to all anchor nodes (i.e known positions in the network relaying on locally available information. Furthermore, inspired by the hunting mechanism of grey wolves, we develop a correction mechanism to further improve localization accuracy without incurring any additional costs. Simulations results show the proposed algorithm achieves better performance and less communication overhead than the most representative localization algorithms currently exist in literature. In addition, real world experiments with MICAz sensor nodes also prove the superiority of the proposed algorithm.**

***Index Terms*—Wireless sensor networks, Localization, Hop-count, MICAz mote, CONTIKI operating system, grey wolf optimizer.**

I. Introduction

Wireless sensor network WSN is a combination of an electrical devices. These electrical devices are microcomputer, transceiver and power source pieces that gives the creativity for WSN. While, the WSN generates an electrical signal to collect the physical data from specific environment. The WSN stores the sensing data in limited memory in microcomputer, then forwards this sensing data to a base station or a central computer by transceiver, both of these electrical devices take a power source from low priced battery. Due to the efficiency of WSN, the WSN inserts in several applications like environmental monitoring, health care, and military surveillance [1]. These applications are requiring the accurate position from where this sensing data transmits, hence, the sensing data is fully meaningless if the location from where the sensing data transmitted is unknown.

Although Global positioning system(GPS) based localization schemes can be used to find a quasi-exact node position, the non-availability of GPS in indoor environment, huge battery consumption and the expensive cost of GPS devices prevent their use in large scale WSNs. So far, several localization algorithms have been proposed in the literature [2]-[5].These algorithms can be roughly classified into three categories range-based, heretical and analytical.

Range based algorithms assume that the distance is measured with respect to a neighbor node by studying a certain characteristics of a signal exchange between the nodes. This distance computes using one of the three methods, Time of arrival (TOA) [6], Angle of arrival (AOA) [7], Received signal strength (RSS) [8]. TOA uses the signal propagation time and known velocity to compute the distance between nodes. But the nodes require high synchronization clocks to determine the signal received time that causes a time delay communication between the nodes. RSS uses a power of the signal and knowledge of the transmitted power as information to obtain the distance. That affects with natural environment by appearing the noise, the reflections and the multipath propagation thereby hindering senor localization accuracy. AOA uses a direction of signal propagation and determining the maximum signal strength during the antenna rotation to compute the distance between nodes. That requires very accurate hardware thereby, adds significant hardware size for WSN. However, range based Range based algorithms are more than accurate heretical and analytical algorithms by avoiding an additional hardware and a high power to ensure the communication between nodes.

Unlike range-based algorithms, a heretical and an analytical algorithms relies to the connectivity information (i.e: hop count) to derive the distance between the nodes. The heretical algorithms such as DV-Hop [2] compute the average hop size distance at anchor nodes and then broadcast it to regular nodes throughout the network which adds a prohibitive overhead to a WSNs. Popular alternative, analytical algorithms are more power efficient than heretical algorithms, where any distance is locally computed at regular nodes which reduces the overhead. However, the traditional distance estimation approaches, a cumulative error is expected to occur when estimating the distance between each regular node-anchor pair, thereby hindering localization accuracy. In this paper new localization algorithm is proposed, that directly estimates the distance estimation.

The rest of this paper is organized as follows: In the following section we give an overview of the network model. In section III the motivation to propose localization algorithm. Section IV provides a detailed description of our proposed

localization algorithm. Section **??** shows simulation and implementation results to characterize the performance of our proposed algorithm. Finally, Section IXconcludes the paper.

II. Network model

A network model contains N sensor nodes deployed in 2-D square area S, the form of deployed is uniform distribution. The disc of N node has the node as an origin point and the transmission capability T_c as a radius. Due to homogenous nature of WSN, nodes are assumed to have same transmission capability. Which, the node is able to communicate directly to the other nodes located inside its disc. Otherwise, by using the multi hop process the node is able to communicate with other nodes who located outside its disc. It also assumed that the sensor nodes separated into two types, the first type called anchor nodes that has a ware location information through compiled the location using manual configuration or GPS. The second type called regular nodes that needs to compute its own location by using the anchor nodes information. As shown in Fig. 1 an anchor nodes have the red triangles shape, the regular ones has the blue circle shape and the dashed line shows the communication between the nodes. In indeed, the ones dashed line is equivalent to one hop.

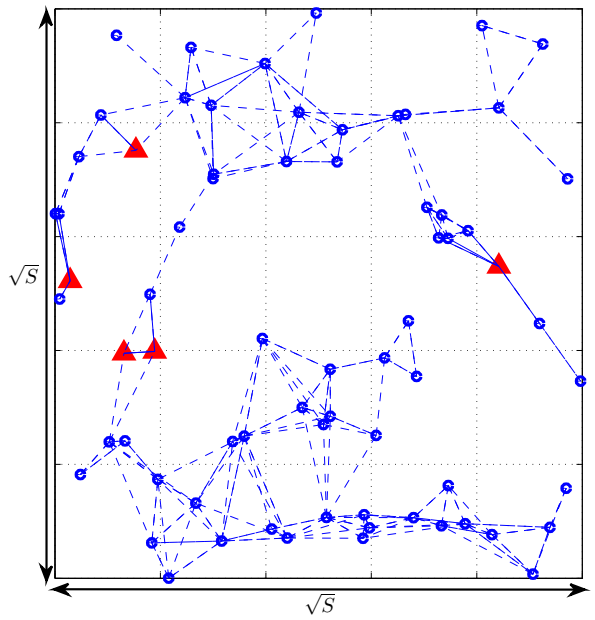

Fig. 1. Network model.

Let N_u and $Nu = N - N_a$ denote the number of anchors and regular nodes. Without loss of generality, let (x_i, y_i), $i = 1, \ldots, N_a$, be the coordinates of the anchor nodes and (x_j, y_j), $i = N_a+1, \ldots, N_a$ be the coordinates of the regular nodes. In what follows the motivation to propose localization algorithm.

III. Related work and motivation

The fundamental requirements to estimate the regular node positrons is the computation of a distances between it and at least three anchor nodes. Then the $k-th$ anchor node broadcasts its location (x_i, y_i), while the hop count recognizes at node by h=1 if the node has immediately communication with $k-th$ anchor. Otherwise, if the node location is far from the directly communication of the $k-th$ anchor node, the node recognizes the hop count by h >1. The distance estimation takes different computation techniques based on the algorithm that used. In DV-hop algorithm the distance estimation is measured by following formula:

$$\hat{d} = n \times \bar{h_s} \tag{1}$$

Where n is the hop count, and h_s is the average hop size. The average hop size is computed on all anchor nodes after the $k-th$ anchor node became aware to the other anchor nodes information as:

$$\bar{h_s} = \frac{1}{N_a(N_a-1)} \sum_{k=1}^{N_a} \sum_{j=1}^{N_a} \frac{d_{k-j}}{n_{k,j}} \tag{2}$$

Where d_{k-j} is the distance between anchor k and anchor j, and $n_{k,j}$ is the value of hop count between the two anchor nodes. Once average hop size calculated, the $k-th$ anchor node broadcasts a package contains this average hop size through all the network nodes. In the end of this step the regular node computes its distance to all anchor nodes by using (1).

In another hand, using analytical approach, the distance between regular-anchor pair is computed by deriving an analytical expression for expected hop progress EHP which is given by the following formula:

$$\bar{h_s} = 2ksin(\theta) \int_0^{r_0} a\ell^2 e^{-k\theta(r_0^2-\ell^2)} \tag{3}$$

However, The heuristic algorithms(i.e DV-Hop) need two processors to estimate the position of regular nodes. The first process is broadcasted the coordinates of anchor nodes and the hop count. The second process is broadcasted each anchor node an average hop size after all anchor nodes became known to each other. That produces a big communication delay, more power consumption and overhead due to the non-localized manner of sensor nodes. The analytical approach require only the first process to estimate the position of regular nodes which reduces the overhead at each anchor node, and the power consumption. In next section proposed a new analytical localization algorithm that reduce greatly the localization error

IV. Proposed algorithm

As any first step in any localization algorithm, the anchor nodes broadcast a package separates into two parts header and data payload. The header part carries the identification ID of anchor node, where, the data payload part carries the anchor node position (x_k, y_k) and a hop count n initialized to one. The node received this message stores it in its database and collaborates its hop count value n_k by the received hop count valuen_k=n, then broadcasts the package after increasing n by 1. The flooding of the same package between the nodes in the network consequently gives high probability to the nodes to receive the same package more than one times. Then the node

checks immediately its database information, If the received hop count n is smaller than stored hop count n_k, the node updates the stored hop count by the received one, then it re-broadcasts the package after incrementing hop count by one. If the received hop count n is greater than the stored hop count n_k the node simply rejects the received package. However, if the nodes obviously unaware the $k-th$ anchor information, it adds one to hop count value n then broadcasts the received package. This mechanism continuously available until all the nodes become aware to all anchor positions and corresponding to find a shortest path to each $k-th$ anchor node.

A. Distance Estimation

In order to localize the i-th regular node (i.e., (N_a+i)-th node), the distances between it and at least 3 anchors are usually required. So far, in most analytical algorithms, the i-th regular node estimates its distance to the k-th anchor $d_{k-(N_a+i)}$ using only the information n_k as

$$\hat{d}_{k-(N_a+i)} = n_k \bar{h}_{\mathrm{s}} \tag{4}$$

where $\bar{h}_{\mathrm{s}}$ is a predefined average hop size.

Let ϵ denotes the estimation error of the distance between the average hop size $\bar{h}_{\mathrm{s}}$ and the real size h_{s}

$$\epsilon_{ik} = \bar{h}_{\mathrm{s}} - h_{\mathrm{s}}, \tag{5}$$

Unfortunately, due to this distance estimation approach, a cumulative error is expected to occur when estimating the distance between each regular node-anchor pair, thereby hindering localization accuracy. In this paper, we propose to directly (i.e., without recurring to the distances between consecutive intermediate nodes) estimate the distance d_{k-i} between the k-th anchor and the i-th regular node. This allows to avoid the cumulative distance estimation error incurred by EHP-based localization algorithms [3]-[5]. In fact, such an error, which has a detrimental effect on localization accuracy, increases with n_k.

Let us denote by X the straight line between the regular node-anchor pair, X can be calculated as following:

$$X = z \times \cos(\theta) \tag{6}$$

where z is the random variable that represents the radius of the sector having the $\star$-th node as a center, $\frac{\pi}{3}$ as an angle.

It is easy to show that

$$\begin{aligned} \mathrm{E}\{X|n_k\} &= \frac{\int_0^{\frac{\pi}{3}} \cos(\theta) \int_{(n_k-1)T_c}^{n_k T_c} z f_Z(z)\, \partial z \partial\theta}{\int_0^{\frac{\pi}{3}} \int_{(n_k-1)T_c}^{n_k T_c} f_Z(z)\, \partial z \partial\theta} \\ &= \frac{3\sqrt{3} \int_{(n_k-1)T_c}^{n_k T_c} z f_Z(z)\, \partial z \partial\theta}{2\pi \int_{(n_k-1)T_c}^{n_k T_c} f_Z(z)\, \partial z \partial\theta} \end{aligned} \tag{7}$$

where $f_Z(z)$ is the probability density function (pdf) of r.v. u.

In order to derive $\mathrm{E}\{X|n_k\}$, we start by deriving the conditional cumulative distribution function (CDF) $F_{Z|n_k}(z|n_k)$

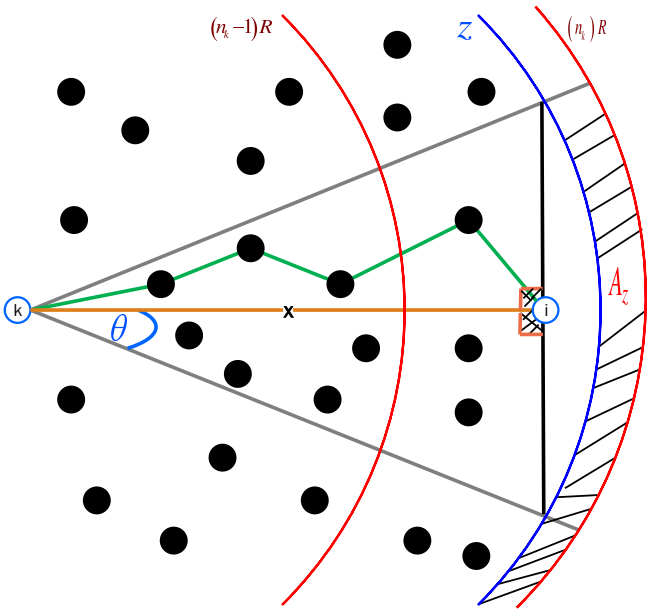

Fig. 2. Distance Estimation.

of Z with respect to n_k. As could be observed from Fig. 2, $Z \leq z$ is guaranteed only if there are no nodes in the area A

$$A_z = S(k, \theta, n_k T_c) - S(k, \theta, z) \tag{8}$$

with $S(\star, \theta, x)$ is the sector having the $\star$-th node as a center, $\frac{\pi}{3}$ as an angle, and x as a radius. $F_{Z|n_k}(z|n_k)$ can be then defined as

$$F_Z(z)\,(z|n_k) = P\,(Z \leq z|n_k) = P\,(\mathrm{E})\,, \tag{9}$$

where $P(\mathrm{E})$ is the probability that the event $\mathrm{E} = \{$no nodes in the area $A_Z\}$ occurs.

Since the nodes are uniformly deployed in the surrounded area (cf. Fig. 1), the probability of having K nodes in A_z follows a Binomial distribution $\mathrm{Bin}\,(N, p)$ where $p = \frac{A_z}{S}$ and S is the total area surface . For relatively large N and small p, it can be readily shown that $\mathrm{Bin}\,(N, p)$ can be accurately approximated by a Poisson distribution $\mathrm{Pois}(\lambda A_z)$ where $\lambda = N/S$ is the average nodes density in the network. Consequently, for a large number of nodes N and small p, we have

$$F_{Z|n_k}(z|n_k) = e^{-\lambda A_z}. \tag{10}$$

It can be readily shown from Fig. 2 that

$$A_z = \frac{\pi}{3}\left(n_k^2 T_c^2 - z^2\right), \tag{11}$$

and, hence,

$$f_Z(z) = \frac{2\lambda\pi}{3} z e^{\frac{-\lambda\pi}{3}\left(n_k^2 T_C^2 - z^2\right)}, \tag{12}$$

By substituting (12) into (7), followed by u-substitution and partial integration, $\mathrm{E}\{X|n_k\}$ is given by (13)

locally available at the i-th regular node and, hence, its computation does not require any additional overhead or power

$$\mathrm{E}\{X|n_k\} = \frac{3e^{-\frac{\lambda\pi}{3}n_k^2T_c^2}\left(2\sqrt{3}\left((n_k-1)e^{\frac{\lambda\pi}{3}(n_k-1)^2T_c^2}-n_ke^{\frac{\lambda\pi}{3}n_k^2T_c^2}\right)\sqrt{\lambda}T_c-3erfi\left(\sqrt{\frac{\lambda\pi}{3}}(n_k-1)T_c\right)+3erfi\left(\sqrt{\frac{\lambda\pi}{3}}n_kT_c\right)\right)}{4\pi\sqrt{\lambda}\left(e^{\frac{1}{3}(1-2n_k)\lambda\pi T_c^2}-1\right)} \quad (13)$$

cost. Such a feature is actually suitable for WSNs wherein power is a scarce resource.

In the same way the nominator integral in (7) can be calculated, and then, the distance d_{k-i} between the k-th anchor and the i-th regular node can be estimated as shown in (13).

It follows (13) that $E\left(Z/n_k\right)$ increases with R and it can be readily computed at each regular node given the a priori knowledge of the node distribution before WSN deployment.

Therefore, the$i$-th regular node is able to obtain an initial estimate of its coordinates (x_0, y_0) using multilateration technique.

B. Virtual hunting-based correction mechanism

Inspired by the social behavior and hierarchy hunting strategy of grey wolves, a new mechanism to adjust the initial coordinates (x_0, y_0) for the regular node(prey) is proposed in this section. The grey wolves hunting strategy is divided into three main steps: tracking/approaching the for prey, encircling prey, and attacking the prey. Based on the hunting strategy the grey wolves are classified into four categories such as alpha(α), beta(β), delta(δ), and omega(γ), where, Alpha wolves are the leader and play a major role in producing new solutions, and Omegas are the less important wolves but, they still help others from facing internal problems. In this paper, α, β, δ, represent the first three nearest anchor nodes to the regular node, where, γ represents all the other anchor nodes as shown in Fig. 3.

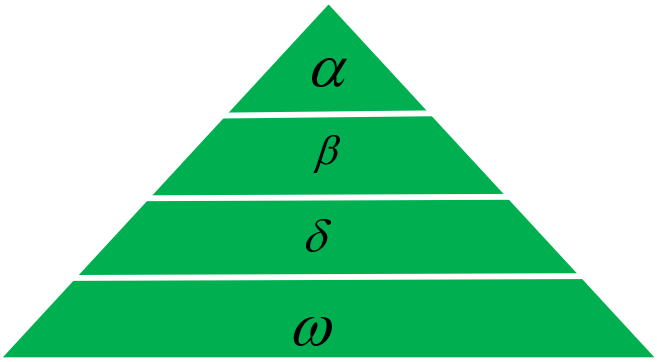

Fig. 3. Gray wolf social hierarchy

In encircling phase, the initial position vector of the regular node is defined, and other search agents adjust its position based on the best solution obtained. The equation of encircling the regular node is given below

$$\begin{aligned}\vec{D} &= \left|\vec{C}\cdot\overrightarrow{X_r}(t)-\vec{X}(t)\right| \\ \vec{X}(t+1) &= \vec{X}_r(t)-\vec{A}.\vec{D}\end{aligned} \quad (14)$$

Where t is the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_r$ is the position vector of the prey, and $\vec{X}$ shows the position vector of a anchor nodes.
The vectors $\vec{A}$ and $\vec{C}$ are computed as follows:

$$\vec{A} = 2\vec{a}\cdot\overrightarrow{r_1}-\vec{a} \quad (15)$$

$$\vec{C} = 2.\overrightarrow{r_2} \quad (16)$$

Where a is a linearly decreased from 2 to 0 over the cruse iteration, and r_1, r_2 are random vectors in $[0, 1]$.

In order to mathematically simulate the hunting behavior of grey wolves, we suppose that the hunting strategy is usually guided by alpha, beta and delta anchor nodes which are considered as the first, second and third best candidate. Therefore, we save the first three best solutions obtained so far and oblige the other search agents including the omegas anchor nodes to update their positions according to the position of the best search agent. In other words alpha, beta, and delta estimate the position of the regular node, and other anchors virtually updates their positions randomly around the regular node. The equations of this mechanism are formulated as follows:

$$\begin{aligned}\overrightarrow{D_\alpha} &= \left|\overrightarrow{C_1}\cdot\overrightarrow{X_\alpha}-\vec{X}\right| \\ \overrightarrow{X_1} &= \overrightarrow{X_\alpha}-\overrightarrow{A_1}\cdot\overrightarrow{D_\alpha} \\ \overrightarrow{D_\beta} &= \left|\overrightarrow{C_2}\cdot\overrightarrow{X_\beta}-\vec{X}\right| \\ \overrightarrow{X_2} &= \overrightarrow{X_\beta}-\overrightarrow{A_2}\cdot\overrightarrow{D_\beta} \\ \overrightarrow{D_\delta} &= \left|\overrightarrow{C_3}\cdot\overrightarrow{X_\delta}-\vec{X}\right| \\ \overrightarrow{X_3} &= \overrightarrow{X_\delta}-\overrightarrow{A_3}\cdot\overrightarrow{D_\delta}\end{aligned} \quad (17)$$

where $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$, $\overrightarrow{D_\delta}$ and are the modified distance vector between the alpha, beta, and delta position to the other omega anchors nodes and $\overrightarrow{C_1}$, $\overrightarrow{C_2}$, $\overrightarrow{C_3}$ and are three coefficient vector aids in adjust distance vector and it is computed using (16).

Finally, the regular position can be updated as following:

$$\vec{X_r}(t+1) = \hat{x}, \hat{y}] = \sum_{k=1}^{3}\frac{\overrightarrow{X_k}}{3} \quad (18)$$

where $\vec{X_r}(t+1)$ is new adjusted new regular position computed by average sum of all positions obtained using alpha beta and delta anchor nodes.

After each iteration, the regular node will compute the following cost function :

$$\Phi(X_r) = \frac{1}{N_a}\sum_{k=1}^{N_a}\frac{\left(\mathrm{E}\left(X|n_k\right)-(\hat{x}-x_k)^2-(\hat{y}-y_k)^2\right)}{n_k^2} \quad (19)$$

According to the calculated cost function $\Phi(X_r)$, the position of alfa, beta and delta will be updated if the value of $\Phi(X_r)$ is less than the previous value as shown in Fig. 4.

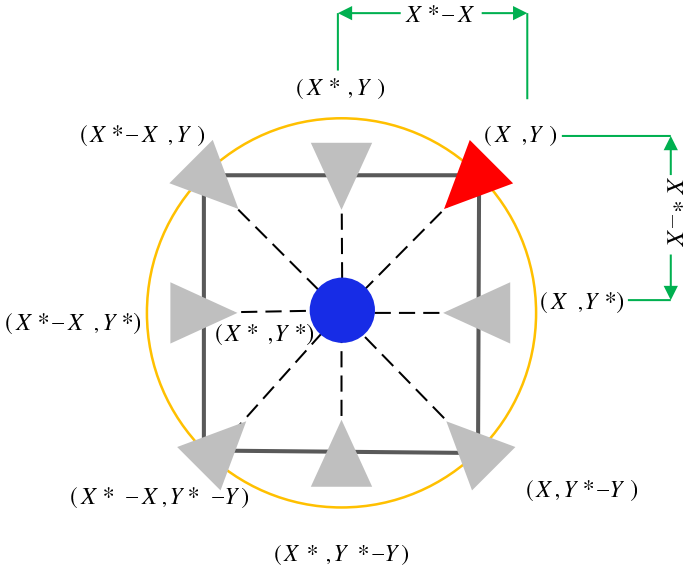

Fig. 4. Updating position of gray wolves in GWO

The over flow of our algorithm is shown in Algorithm. 1

Algorithm 1 over flow for Localization algorithm

Input Number of anchor Na and their positions (x_k, y_k). Where k=1,2...,Na.
for (i=1 $\rightarrow$ Na) **do**
  $\hat{d}_{k\rightarrow i}$ Using equation (13)
end for
$x_0, y_0 \leftarrow$ Multilateration
Initialize $X_{(\alpha)} \leftarrow x_1, y_1$ the first best solution
Initialize $X_{(\beta)} \leftarrow x_2, y_2$ the second best solution
Initialize $X_{(\delta)} \leftarrow x_3, y_3$ the third best solution
$\Phi(X_r) \leftarrow$ Using equation (19) % fitness of each search agent%
while ($k < \varepsilon$) **do**
  for (i=1 $\rightarrow$ Na) **do**
    Update the current search agent position
  end for
  $\Phi(X_r) \leftarrow$ Using equation (19)
  Update the coefficient of the vectors a,A and C
  if ($\Phi^k(X_r) < \Phi^{k-1}(X_r)$) **then**
    Update the best agent $X_{(\alpha)}$,$X_{(\beta)}$, $X_{(\delta)}$
    k=k+1
  end if
  X_1, X_2, X_3 Using equation (17)
end while
$(\hat{x}, \hat{y})$ Using equation (18)

V. Hardware And Software Requirements

A. Micaz Mote

MPR2400 MICAZ mote WSN is designed for deeply embedded sensor networks. While MICAZ offers a high data rate ratio 250kbps donated by using zigbee-ready radio. The zigbee-ready radio 2.4 GHz band is compliant RF transceiver with IEEE 802.15.4 and It runs the Moteworks from its internal flash memory [9] based on low power microcontroller ATmega 128L. The 51 pin expansion connector allows the MICAZ mote to connect with extremely peripherals (ex: data acquisition board) and the both analog and digital input/output can manage the 51 pin instruction. The MICAZ mote offers a hardware security (AES-128). Furthermore, MICAZ mote can be exploited to run sensor application and the network communication simultaneously.

B. MIB520CB Mote INTERFACE BOARD

The connection between MIB520CB mote and MICAZ mote is the 51-pin expansion connector. The interface board controls the MICAZ mote communication and programing by identification of USB. Through the USB connection between the interface board and PC offers for interface board a two com port, an input port works for in-system mote programming and an output port works for data communication over USB [10]. Note that the MIB520CB mote interface board should install FTDI FT2232C to use USB port as a virtual COM port.

C. Contiki Operating System

MoteWorks- optimizes a low battery-operated networks and provides end to end solution across all tiers warless sensor networking application. MoteWorks includes an open source operating system that takes several forms that should be chosen based on application requirements. In Some applications the high importance requirement that the nodes can operate unattended for very long period of time, in contract another applications the high importance requirement that the nodes should be able to process a huge amounts of information for short term. In this paper we gave the priority to CONTIKI operating system instead of Tinyos, after made comparison [**?**] between Tinyos and CONTIKI operating systems. However, the Tinyos is better to use when the resources are limited and every little bit of saved memory. In other hand CONTIKI is more flexible when the node software should update often for a high density of nodes.
The Contiki Operating System simulation considering as: When simulating the node native code, the node type actives as a link between a node and the compiled Contiki system. While only the node type has functions able to interact with a loaded Contiki system such as functions of copying and replacing memory assignments, initializing the Contiki system and finally a function that notices the system to handle an event as shown in Fig. 5.
When the sensor node is running, its information memory is copied into the C environment and a Java Native Interface (JNI) is calling to make the event driven Contiki Kernel. In indeed, for each loaded Contiki system COOJA should be find the address of functions and variables. This issue solves by parsing the map file generated at link-time. The map file consists of information about symbols addresses in the library.

Then the Cooja is able to find the addresses of all the library variables by comparing the absolute memory address of a variable at runtime with its relative address specified in the map file.

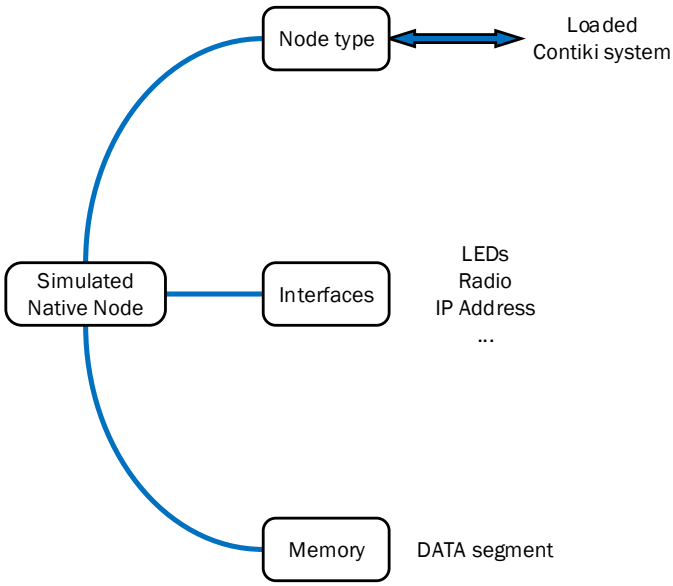

Fig. 5. Node Type Acts As Link between loaded .

VI. Setup The Real World Implementation

This section presents how to setup both software and hardware equipment's for apply localization algorithm into WSN in details.

A. Hardware setup

First to compile the executable file to MICAZ mote, the MICAZ mote should be connected to the PC through the USB. Second is establishing a base station or the getway that will be receiving the transmission information from the nodes and displays the received data on the PC. As shown in Figure 6 note that the base station is constructed without battery, due

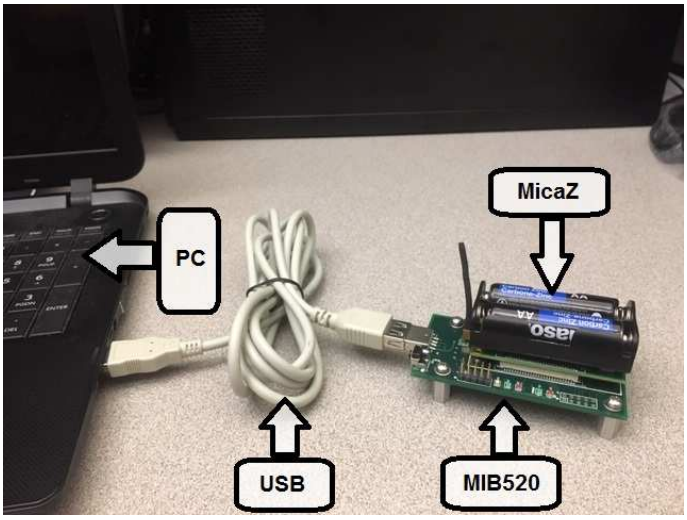

Fig. 6. Connection between MICAz and MIB520.

to the offer of the interface board that makes the MICAz mote ON during the connection to its.

B. Uploading Executable File Into The Mote

After implementing the Hardware setup, the mote needs suitable pre-compile as one of Moteworks [**?**] components like Programmers Notepad 2. Now by creating a networks that contains specific number of nodes like (anchor and regular nodes) in the Contiki COOJA Simulator. The latter offers a Reload button by pressing it the executable file for each node generates in C environment before collaborate the C code of the node by specified the identification ID and the transmit TX. While using the C library (node-id.h) offers a control to specify the ID of the node through flexible function node-id-burn(), and using C library (dev/cc2420.h) offers a control to specify the TX of the node through cc2420-set-txpower() function. Note that the cc2420-set-txpower() function uses dBm power level [**?**] to adjust the TX of the node. Further, the difference between the normal motes and the base station mote is the ID. We assume that the base station is the mote how holds ID equal to one. For the normal motes the ID must be bigger than one. Also to make the base station carries two phases the first phase is considering as normal mote and second phase is the base station.
Now by creating the executable file in Contiki COOJA simulator and adjusting both (ID and TX) to each node the executable files should be transferred to MoteWorks file environments like (to one of application directory (ex: Blink.nc) that attached in Programmers Notepad 2). The directory path in our study is $C : /Memsic/cygwin/opt/MoteWorks/apps/general/Blink/build/micaz$.

Note that executable file in MoteWorks environment renames it to main.exe. The Programmers Notepad 2 allows to manage the shell by clicking on tools-¿shell to fill the edit text by general syntax for installing that is :
"make $platform$ re/install, n $programmer$, $port$"
Where $platform$ is a Mote platform name, n is an optional number (decimal number) to ensure the ID of the motes comparison to the ID in CONTIKI firmware, $programmer$ is the name of the mote equipment is used, $port$ is the COM Port number that allows to compile the firmware into the mote. The steps to know the available COM port number assignment is given by [**?**]. In our study, COM port number assigned are COM3 and COM4. Keep in mind that the lower number COM port is input like Mote programing and the higher one is output like Mote communication (the input is the COM3 in order to compile the firmware into the mote and the output is the COM4 in order to read the data from the base station and display it on the PC). Hence, the COM Port chosen at this case is the COM3 in order to compile the firmware into the mote. The difference between the install and the reinstall is that, the install comment programs the Mote device, which reinstall comment downloads the pre-compiled program into the Mote device. The latter is significantly faster. The final shape of syntax that we used to upload firmware into the mote is:
"make $micaz$ reinstall,n $mib520$, $com3$".

The layer that used in our study id the Rime layer, that offers in Contiki operating system based on his advantages. The Rime layer is a lightweight layered communication stack [**?**] for sensor network and is design to be much simpler than existing protocols for modular communication abstractions for sensor networks. Further, the Rime layer is considering as an intermediate layer between the effort of local neighbor broadcast and reliable neighbor unicast and the effort network flooding and hop-by-hop reliable multi-hop unicast. However, Our application form is low power mode due to reduce the power consumption by sleep the process in case no action to do.

C. Software Setup

Due to our requirements to collect the information of all nodes exit in the network we offered a GUI interface user as shown in Fig. 8. That has the ability to collect the sensors data through stender communication interface USB. The USB parameter's selected by the user in order to choose a specific COM Port that fits their case. In our study the COM4 is the chosen port to read data.

VII. Testing And Debugging The Real World Implementation

WSN applications have been implemented at many places and taking different tasks for example in Columbia the WSN detects and prevents forest fire [**?**] and in india the WSN modified to be useful for disaster management purposes [**?**]. Both of the previous examples are messing the information of sensing data position. Furthermore, in our Wireless lab (Canada) we modified the WSN to improve the efficiency of the localization algorithms in our literature [**?**] in real world. This section shows the application WSN in real world to obtain the location of MICAZ motes who are regular node. Then declares these location over graphic 2-D. finally we mention the debugging method used and some instructions should be verified to do implementation for WSN in real world. The proposed algorithm and the best representative localization algorithms (like DV-Hop and LEAP) are tested over WSN's using the parameters show in Table I.

TABLE I
The Hardware Implementation Parameters.

Name	Value
Dimension	2-D
Anchor number	3
Regular number	20
Area	80cm*80cm
Transmission capability	10cm

The WSN implementation area took place on a grid contains of 2-D coordinate system as shown in Fig. 7. Where the MICAZ mote with Blue tickets are anchor nodes and the MICAZ motes with Yellow tickets are the regular nodes. Where the MICAZ motes plot over the grid by its location (X,Y) and the transmission capability is the same to each MICAZ mote in the network.

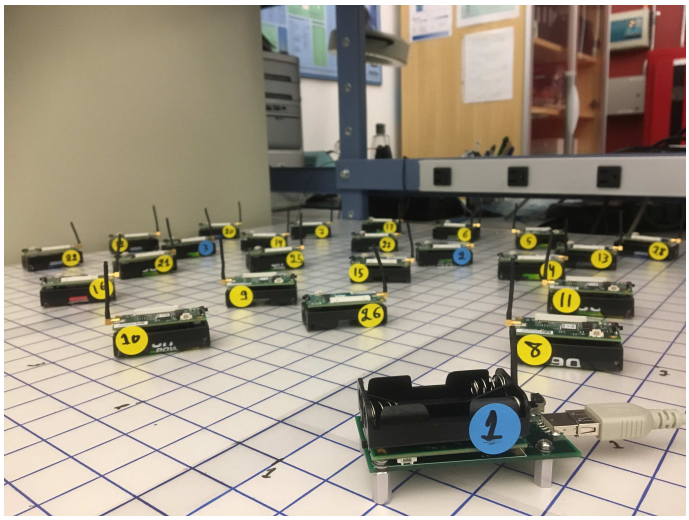

Fig. 7. Hardware Implementation.

The first step of our proposed algorithm the anchor nodes broadcast their packet through the network. In our WSN application this step is implemented in individual way, where one anchor just broadcasts its packet information through the network until anther anchor gets permission from the sender anchor to broadcast. The packet information contains the identification of anchor node (ex: ID) and the coordinates of anchor node (ex: (X,Y)). Note that the permission for broadcast takes incrementing way for anchor ID. When the node received the anchor node packet the node stores this packet in its memory, then increments the hop count by one after that broadcasts this packet to the neighbor how locate in its transmission capability range. Once the node received this packet again, the node checks its memory, if the received hop count is less than the stored memory hop count the node replaces the received packet instead of the stored one as shown in Fig. 8 the node with (ID 1) did correction by notification the following information (the received hop-count is less than the stored one. Do correction). Where the stored packet of anchor ID 3 the hop count was 7 but the received packet holds hop count 3 from the anchor 3, then the stored hop count greater than the received hop count (7>3) so updates the packet information in memory then broadcasts this packet after implement hop count by one. Else If the received information is greater than the stored one the node killed this packet. However, if the node is unknown to the anchor node packet information it stores this packet information in its memory then broadcasts after incremented hop count by one as shown in Fig. 8 (ex: the anchor (ID 2) is received by hops=1 at anchor (ID 1)) . This mechanism is available until all the regular nodes find the shortest path to all anchor nodes relaying on minimum hop count value to each anchor. After received the all anchor nodes information the node is able to compute its estimation location ((X,Y). As shown in Fig. 8 all the regular node estimated their positions (ex: Node-ID 8, x=19.71,y=46.03).

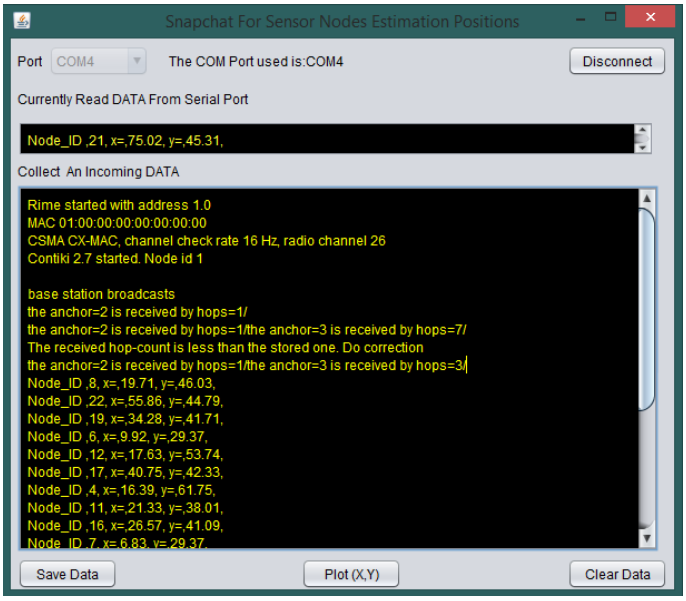

Fig. 8. Network Information

After all the communication information between the nodes in the network stopped the GUI displays all regular node estimation coordinates.Then a user is able to see the error between the real coordinates and the estimate one by pressing Plot(X,Y) button that shown in Fig. 8 . the Fig. 9 shows the real location of regular nodes by green circle shapes, the red square shapes represent the estimate location and the dashed line is a link between the both shapes green circle and red square that is considering as the error evaluation. Further, by pressing the Clear Data button does reset to the network stored information.

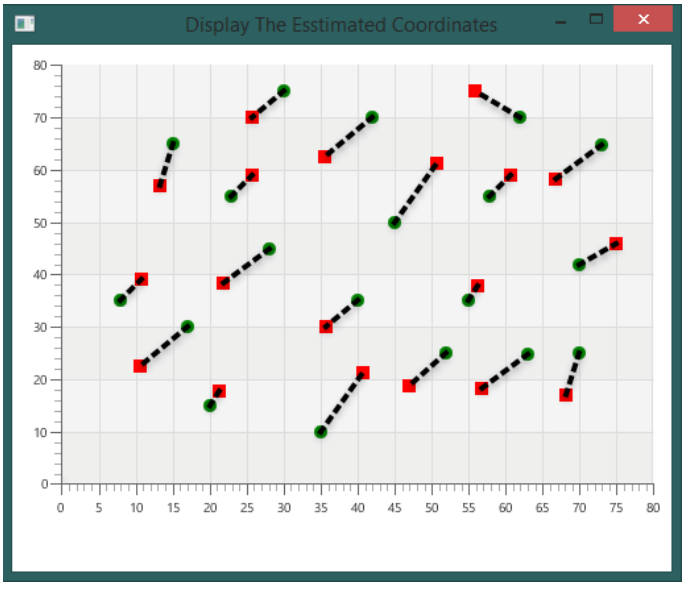

Fig. 9. Display a regular node location in cartesian coordinate system.

As we can see in Fig. 10 our proposed algorithm outperforms DV-Hop and LAEP in a real-world deployment. As expected, the NRMSE acheived by our proposed algorithm more accurate than the DV-Hop and the LAEP, and decreases with increasing the number of the anchor nodes in the network. This result obtained over 20 iterations at each specific number of anchor.

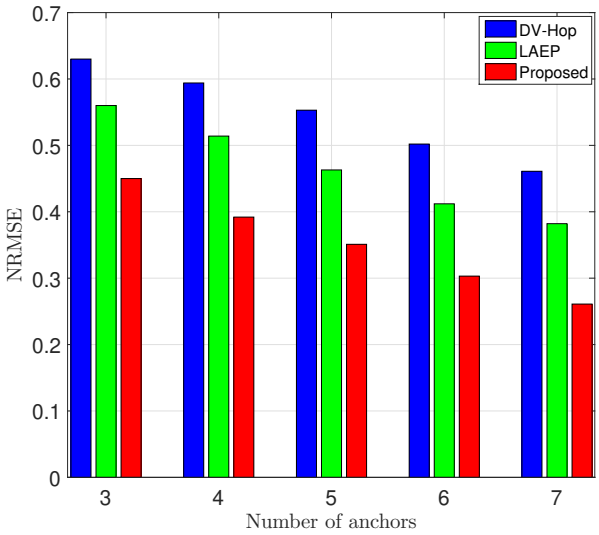

Fig. 10. Indoor evaluation: NRMSE versus number of anchors

In indeed, the communication between the nodes takes two form. First form is broadcast communication protocol where all the anchor nodes broadcast their packet information through the network and the same broadcast protocol is applied in all regular nodes. After regular nodes compute their location by using Multeritation method the all nodes in the network is applied the second form to send this estimation coordinates that is Multicast communication protocol.

The Multicast communication protocol is a group communication and considers as many to many distribution. While the transmission of the packet should be addressed to specific node in the network for example in our prosed algorithm the transmission of the packet is addressed to the node who has identification ID is 1, while during the first broadcast of the anchor node ID 1 all nodes maintained this node identification (EX:ID=1) in their memory. Note that in our network the anchor node with ID 1 is the base station. However, the advantage of using Multicast communication protocol is reduced the power consumption.

To avoid the network destroyed or any error in WSN real world implementation the following structure should be verified:

1) use the LED's to do hardware debugging, in our work we used the LED's as shown in Table II below:

TABLE II
HARDWARE DEBUGGING

YELLOW led	used to show the preparation of anchor nodes to broadcast their packages
GREEN led	used to show if the package received
RED led	used it to see if the regular nodes received information of all anchor node in the network

2) the base station is established by mounting one MICAz mote excluding the power source (battery's) to the MIB520 mote interface board. Furthermore, the base station must be connected to the PC via USB port.
3) During the process compiling the firmware into the mote, choosing a correct pre-compiler that depends on the system requirement.
4) Before the data can be showed, each node and the base station that appears in GUI, all of them have to be configured in individual way. further to the collaboration of ID and TX to each node should be classified.
5) To get accuracy reading from the sensor, the sensor might be fixed at the ground. This is because the strategy of our algorithm implemented in 2D dimension to obtain accurate localization. So, if the sensor is not completely fixed, it will pick up false reading, causing far localization estimate comparison to accurate one.
6) the batteries have to be powered to do successfully signal in the network. Otherwise the WSN has no ability to transmit a signal to other WSN neighbors.

VIII. STIMULATION

In this section, the performance of the proposed algorithm is evaluated by establishing a simulator in Matlab. The performance baseline is stated by comparison under the same setting of the proposed algorithm versus the best representative localization algorithms mentioned in the literature, i.e.., DV-Hop, LEAP and EPHP. In all stimulations all the nodes are uniformly deployed in a 2-D square area S. The transmission capability is equivalent across the network is equal at all nodes. We always assume that the amount of anchor nodes Na is fixed over all iterations and that the total number of regular nodes N is exchangeable from one stimulation to another. the stimulation results are obtained by averaging over 100 trails.Table III is shown the all stimulation parameters:

TABLE III
SIMULATION TABLE

Name	Value
Anchor density	20
Regular density	400..700
Area	100m*100m
Transmission capability	10cm

To evaluate the proposed algorithm performance, one metric is utilized, which is normalized root mean square error (NRMSE), that defined as follow:

$$NRMAE = \frac{1}{N_u}\sum_{i=1}^{N_u}\frac{\sqrt{(x_i-\hat{x_i})^2+(y_i-\hat{y_i})^2}}{T_{c_i}}. \quad (20)$$

Fig. 11 shows the localization NRMSE for different anchor's percentage with constant of regular node amount equal to 200. Our proposed algorithm achieves a good performance by getting lowest localization NRMSE curve comparison to the other DV-Hop and LEAP. However, that was expected when increasing the anchor's node in the network the accuracy should be high.

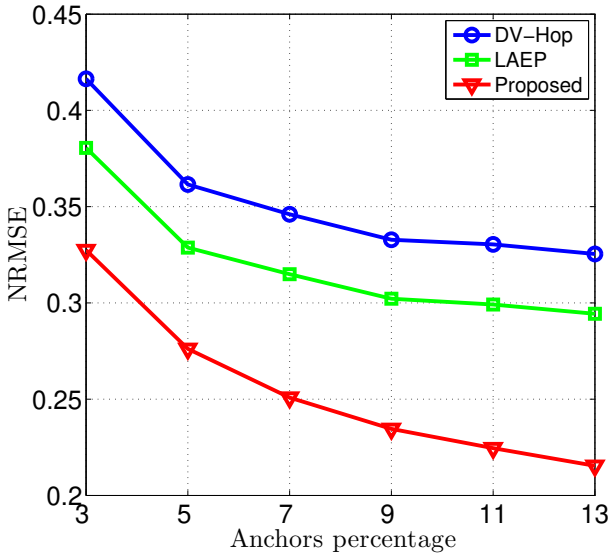

Fig. 11. NRMSE Versus Anchor Percentage.

Fig. 12 shows the NRMES achieved by DV-Hop, LEAP and the proposed algorithm for different node density with a constant anchor's percentage of 10%. As shown in Fig. 12 the proposed algorithm is until three and four times accurate than LEAP and DV-Hop, while our proposed algorithm achieved a slightly decrease when the node density increase.

Fig. 13 plots the localization NRMSEs achieved by DV-Hop, LEAP and proposed algorithm for communication range. As can be seen form this figure, the proposed algorithm achieves high accuracy with increasing the communication range more than the DV-Hop and LEAP. That because the increasing of communication range reduces the hop count.

IX. CONCLUSION

In this paper, a novel localization algorithm tailored for WSN is developed where each regular node will locally estimate its distance to all anchor nodes in the network relaying only on available information. Furthermore, a correction mechanism to further improve localization accuracy inspired by the hunting mechanism of grey wolves is developed. The stimulations and implementation show that the proposed algorithm outperforms the localization algorithms that exist in the literature in terms of accuracy.

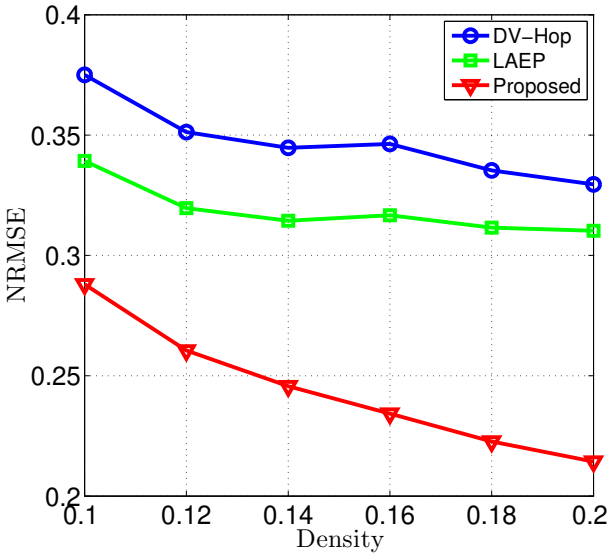

Fig. 12. NRMSE Versus Anchor Density.

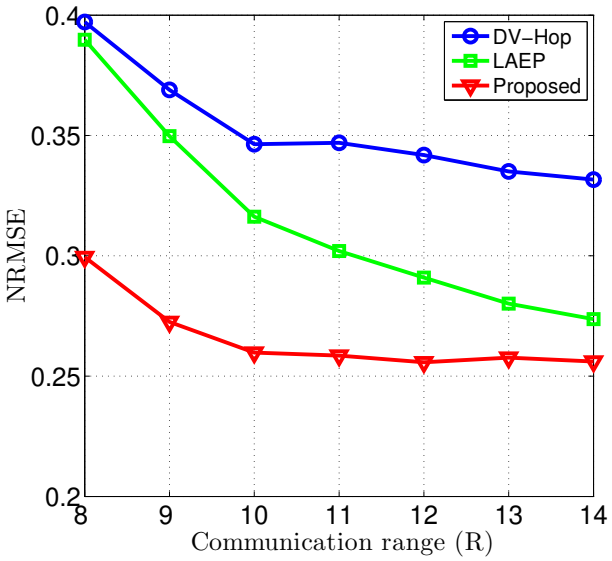

Fig. 13. NRMSE Versus Communication range.

REFERENCES

[1] F. Gustafsson and F. Gunnarsson, "Mobile Positioning Using Wireless Networks: Possibilities and Fundamental Limitations Based on Available Wireless Network Measurements," IEEE Signal Process. Mag., vol. 22, no. 4, pp. 41-53, July 2005.

[2] D. Niculescu and B. Nath, "Ad hoc Positioning System (APS)," *Proc. IEEE GLOBECOM,* San Antonio, USA, November 2001.

[3] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, "Range-Free Localization using Expected Hop Progress in Wireless Sensor Networks," *IEEE Trans. Parallel Distrib. Syst.,* vol. 25, no. 10, pp. 1540-1552, October 2009.

[4] B. Huang, C. Yu, B.D.O. Anderson, and G. Mao, "Estimating Distances via Connectivity in Wireless Sensor Networks," *Wirel. Commun. Mob. Com.,* vol. 14, no. 5, pp. 541-556, April 2014.

[5] S. Lee, B. Koo, and S. Kim, "RAPS: Reliable Anchor Pair Selection for Range-Free Localization in Anisotropic Networks," *IEEE Commun. Lett.,* vol. 18, no. 8, pp. 1403-1406, July 2014.

[6] S. Hong, D. Zhi, S. Dasgupta, and Z. Chunming, "Multiple Source Localization in Wireless Sensor Networks Based on Time of Arrival Measurement," IEEE Trans. Signal Process., vol. 62, no. 8, pp. 1938-1949, February 2014.

[7] J. Rezazadeh, M. Moradi, A.S. Ismail and E. Dutkiewicz, "Superior Path Planning Mechanism for Mobile Beacon-Assisted Localization in Wireless Sensor Networks," IEEE Sensors J., vol. 14, no. 9, pp. 3052-3064, May 2014.

[8] H. Ren and M. Q-H. Meng, "Power Adaptive Localization Algorithm for Wireless Sensor Networks Using Particle Filter," *IEEE Trans. Veh Techno.,* vol. 58, no. 5, pp. 2498-2508, June 2009.

[9] "MICAz datasheet". Document Part Number: 6020-0060-04 Rev A.

[10] Interface Board "MIB520 USB Interface Board" datasheet. Document Part Number: 6020-0091-04 Rev A.

[11] S. Goyal and M. S. Patterh, "Modified bat algorithm for localization of wireless sensor network," Wireless Personal Communications, vol. 86, pp. 657–670, 2015.