AN INTEGRATED SOFTWARE FRAMEWORK FOR LOCALIZATION IN WIRELESS SENSOR NETWORK

Michał MARKS, Ewa NIEWIADOMSKA-SZYNKIEWICZ

Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska 15/19
00-665 Warsaw, Poland
&
Research and Academic Computer Network (NASK)
Wawozowa 18, 02-796 Warsaw, Poland
e-mail: {mmarks, ens}@ia.pw.edu.pl

Joanna KOŁODZIEJ

Institute of Computer Science
Cracow University of Technology
Warszawska 24, 31-155 Cracow, Poland
e-mail: jokolodziej@pk.edu.pl

Abstract. Devices that form a wireless sensor network (WSN) system are usually remotely deployed in large numbers in a sensing field. WSNs have enabled numerous applications, in which location awareness is usually required. Therefore, numerous localization systems are provided to assign geographic coordinates to each node in a network. In this paper, we describe and evaluate an integrated software framework WSNLS (Wireless Sensor Network Localization System) that provides tools for network nodes localization and the environment for tuning and testing various localization schemes. Simulation experiments can be performed on parallel and multi-core computers or computer clusters. The main component of the WSNLS framework is the library of solvers for calculating the geographic coordinates of nodes in a network. Our original solution implemented in WSNLS is the localization system that combines simple geometry of triangles and stochastic opti-

mization to determine the position of nodes with unknown location in the sensing field. We describe and discuss the performance of our system due to the accuracy of location estimation and computation time. Numerical results presented in the paper confirm that our hybrid scheme gives accurate location estimates of network nodes in sensible computing time, and the WSNLS framework can be successfully used for efficient tuning and verification of different localization techniques.

Keywords: Localization, wireless sensor networks, stochastic optimization, simulated annealing, genetic algorithm, parallel programming, HPC, software framework

1 INTRODUCTION

Wireless sensor network (WSN) is a distributed system typically composed of small-size, embedded devices grouped into network and deployed densely over a significant area. Each device can run applications and participate in transferring data to recipients within its range. Most applications of WSN require the correlation of sensor readings with physical locations [11, 12, 21]. Wireless sensor network localization can be solved in different ways [10]. Many automated localization systems for assigning geographic coordinates to each node have been designed and developed [1, 2, 11, 12, 17, 21]. All these schemes should work with inexpensive off-the-shelf hardware, minimal energy requirements, scale to large networks, achieve good localization accuracy in the presence of irregularities and give the solution in the short time. The development of new schemes for nodes localization is an important direction of research in the field of modern ad hoc networks.

The localization techniques can be classified with respect to various criteria. They differ in assumed localization precision, hardware capabilities, measurement and calculation methods, computing organization, assumed network configuration, nodes properties and deployment, etc. [15]. Numerous techniques consist in identification of approximate location of nodes based on merely partial information on the location of the set of nodes with known positions in a sensor network and measurements of inter-node distances. Different methods can be used to estimate inter-node distances [1, 2, 10, 15]. With regard to hardware capabilities of devices that form a network, and mechanisms used for the computing inter-node distances, we divide the localization methods into two categories:

- distance based (range based) methods,
- connectivity based (range free) methods.

The former methods are defined by schemes that use absolute point to point distance estimates or angle estimates in location calculation. The latter methods make no assumption about the availability or validity of such information, and use only information about connections between nodes to locate the entire sensor network. The popular connectivity based solutions are hop-counting techniques. Distance

based methods usually require an additional equipment allowing much better resolution than in case of connectivity based ones. Therefore, in our research we focus on distance based methods.

Evaluation of localization methods can be performed using analytical techniques, computer simulation and practical evaluation in a WSN testbed. The complexity of many WSN systems often makes analytical methods unsuitable. The simulation is a relatively easy and highly available technique to validate and compare the performance of localization schemes and to tune their parameters before the application in a real network. Unfortunately, the performance of the distance based localization system strongly depends on the accuracy of inter-node distances estimation. Moreover, it was observed [16] that different methods are suitable for different network sizes and topologies. Therefore, development of high accuracy localization system that can operate in various networks requires multiple execution of this system for different values of parameters specific to the considered algorithms and various test networks. In general, WSN localization is a complex task and usually involves cumbersome calculations, especially when considering large scale networks with hundreds or thousands of nodes. The direction which should bring benefit is parallel computing where multiple simulations are carried out on different cores, processors or computers.

We have designed and developed an integrated software framework that provides tools for localization of devices that form various types of WSN. Our system can operate in two modes. It can be used to estimate the geographic coordinates to each node with unknown position in the real life sensor network (on-line mode). Moreover, it can be used for performance evaluation of various localization systems that are integrated with the framework (off-line mode). High Performance Computing (HPC) techniques are employed to speed up calculations.

The rest of this paper is organized as follows. Two stages of the distance based localization system operation are described in Section 2. Next, we provide an overview of our WSNLS framework in Section 3. We focus on localization solvers. Finally, in Section 4 we present and discuss the numerical results of our localization systems and show the effectiveness of the WSNLS framework. The paper concludes in Section 5.

2 DISTANCE BASED LOCALIZATION

Let us consider a network formed by M sensor devices (anchor nodes) that are aware of their location, either through GPS or manual recording and entering position during deployment, and N sensor devices (non-anchor nodes) that are not aware of their location in a network system. The goal of a localization system is to estimate coordinate vectors of all N non-anchor nodes. In general, distance based localization schemes operate in two stages:

Inter-node distances estimation stage – estimation of true inter-node distances based on inter-node transmissions and measurements.

Position calculation stage – transformation of calculated distances into geographic coordinates of nodes forming the network.

2.1 Inter-Node Distances Estimation Stage

In accordance with the available hardware distance based localization systems exploit the following techniques widely described in literature [5, 10, 12]: Angle of Arrival (AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Received Signal Strength Indicator (RSSI). AoA, ToA and TDoA methods need an additional stuff such as antennas or accurately synchronized clocks. The most popular is the RSSI method due to low cost (no additional hardware), easy configuration and deployment. The disadvantage of this solution is low level of measurement accuracy because of high variability of RSSI value [5, 18]. Nevertheless, some authors indicate that new radio transceivers can give RSSI measurements good enough to be a reasonable distance estimator [4, 20].

2.2 Position Calculation Stage

In the position calculation stage the computed inter-node distances are used to estimate the geographic coordinates of all non-anchor nodes in a given network. Position estimation can be done by using different techniques. The widely used ones include triangulation, trilateration, multitrilateration and multidimensional scaling. The common idea of other methods is formulating the localization problem as the linear, quadratic or nonconvex nonlinear optimization task solved by linear, quadratic or nonlinear (often heuristic) solvers. Recently, a popular group consists of hybrid systems that use more than one technique to estimate location, i.e., results of initial localization are refined using another localization method. All mentioned methods are described and evaluated in literature [1, 6, 8, 9, 11, 12, 14, 17].

3 SOFTWARE FRAMEWORK WSNLS

3.1 Composition and Implementation

The Wireless Sensor Network Localization System (WSNLS) is an integrated software framework that provides solvers for localization of nodes of WSN and the environment for testing various localization schemes on parallel computes or computer clusters. Due to open architecture and object-oriented programming the software can be easily extended with implementations of new approaches for calculating positions of nodes in a network. WSNLS can be used to estimate the geographic coordinates to all devices forming the real life sensor network. Moreover, it can be used for tuning and performance evaluation of various localization solvers that are integrated with the framework before their practical application to a real life network.

WSNLS is composed of a runtime platform (formed by two components, i.e., *Distributed Computing Manager* and *Computational Server*) responsible for calculation management and interprocess communication, *Networks Generator* – a component for modeling a network to be simulated, two components responsible for location calculations, i.e., *Distance Estimation Module* and *Position Calculation Module*, database for recording data of all examined networks and results of calculations, and a set of tools to support the interaction with a user (GUI). The architecture of WSNLS is presented in Figure 1, and all components are described in detail in the following subsections.
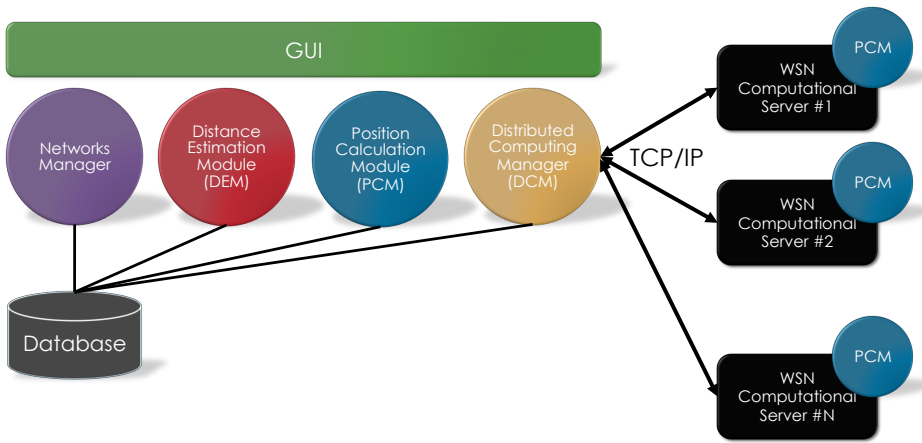

Figure 1. The components of the WSNL system

To speed up the preliminary tuning of a localization solver to a given WSN application, and testing efficiency and robustness of various localization systems our software framework provides the environment for parallel calculations. The highest computational burden is concerned with the estimation of the geographic coordinates of network nodes. Moreover, to tune parameters of the localization solvers or compare the results of localization given by different algorithms we often have to perform multiple simulation experiments for various network topologies and initial parameters. Therefore, many instances of *Position Calculation Module* can be executed in parallel. The other components, i.e., *Networks Generator* and *Distance Estimation Module* are performed in the sequential manner.

3.2 Networks Generator

The *Networks Generator* (*NG*) (Figure 2) provides an interface for low-power networks modeling. User can add, remove and modify networks by selecting appropriate topology, channel and radio parameters. In general, modeling of low-power links is very difficult since the link characteristic depends on a radio chip (e.g., TR1000,

CC1000, CC2420, etc), environment (indoor, outdoor) and many other parameters such as traffic load or radio channel [3]. In our software the models based on *Link Layer Model for MATLAB* described in [22] are provided. The *Networks Generator* focuses on wireless channel modeling, no radio modulation and encoding are considered.
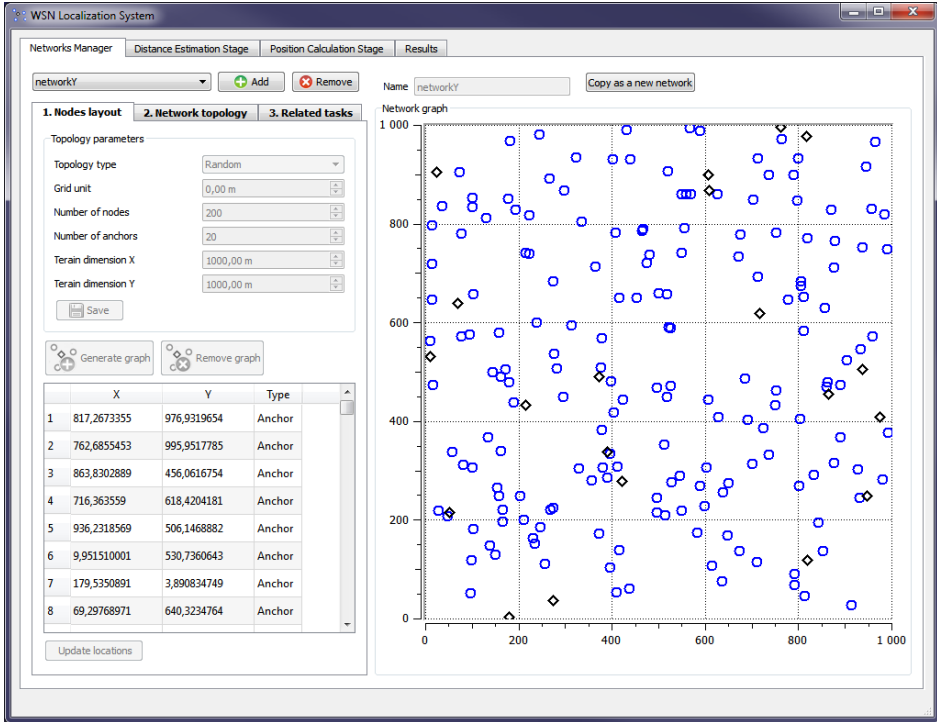

Figure 2. Networks generator in the WSNLS system

3.3 Distance Estimation Module

The *Distance Estimation Module* (*DEM*) provides an algorithm transforming RSSI measurements into estimates of the inter-node distances. The commonly used radio signal propagation model outlined in [13, 19] is used to calculate these distances. Let us define the power of the signal received by a receiver P^r at a distance d:

$$P^r(d)[dBm] = P^t[dBm] - PL(d)[dB], \tag{1}$$

where P^t denotes power used by a sender to transmit the signal and $PL(d)$ the average signal degradation (path loss) with the distance d. The average path loss

$PL(d)$ in (1) is modeled as a function of a distance d raised to an attenuation constant n that indicates the rate, at which the path loss increases with a distance:

$$PL(d)[dB] = PL(d_0)[dB] + 10n \log \left(\frac{d}{d_0} \right),\qquad(2)$$

where d_0 is a close-in reference distance (for IEEE 802.15.4 usually $d_0 = 1\,\mathrm{m}$). Equation (2) was developed as a combination of analytical and empirical methods. Equations (1) and (2) can be used to estimate the average distance $\tilde{d}_{ij}$ between each pair of nodes (i, j) in a network. The estimate of each inter-node distance is a function of the received signal strength:

$$\tilde{d}_{ij} = d_0 \cdot 10^{\frac{P^t - PL(d_0)}{10n}} \cdot 10^{-\frac{P^r_{ij}}{10n}} = \eta \cdot 10^{\theta \cdot P^r_{ij}},\qquad(3)$$

where $PL(d_0)$ denotes the path loss at the reference distance d_0, P^t the output power of the transmitter, P^r_{ij} the received signal strength measured for each pair (i, j) of nodes, $\eta = d_0 \cdot 10^{\frac{P^t - PL(d_0)}{10n}}$ and $\theta = -\frac{1}{10n}$.

Using RSSI measurements and estimated distances between anchor nodes we can calculate the distances between each pair of non-anchor nodes and distances between each pair consisting of anchor and non-anchor nodes. The *DEM* module implements the following algorithm for inter-node distances calculation:

Step 1: Measure RSSI for all pairs of nodes in a network.

Step 2: Compute values of parameters η and θ in (3) solving a least square problem for all pairs of anchor nodes. Three variants of least square problem formulation are provided in *DEM*: Ordinary Least Square problem (OLS), Weighted Least Square problem (WLS) and Geometric Combined Least Square problem (GCLS). The user has to select the best algorithm for his application.

Step 3: Calculate the average distances $\tilde{d}$ to all nodes located within transmission ranges using formula (3) and optimal values of η and θ calculated in step 2.

As mentioned in step 2 of the algorithm three approaches to parameters estimation, i.e., OLS, WLS and GCLS are provided. They differ in the accuracy of the parameters estimation and computing time. The detailed description of these methods together with a performance evaluation for different network topologies are provided in [13].

3.4 Position Calculation Module

The *Position Calculation Module* (*PCM*) is the main component of our framework; it provides methods for estimating the geographic coordinates of non-anchor nodes in the network using the inter-node distances calculated by the *DEM* component. *PCM* implements localization schemes based on geometrical techniques: trilateration and iterative multitrilateration, nonconvex optimization: simulated annealing (SA) and

genetic algorithm (GA), and hybrid technique combining trilateration and stochastic optimization – two versions: TSA and TGA algorithms.

In case of all listed approaches to network nodes localization the mathematical model of the WSN localization problem is formulated. The trilateration technique is a simple method of determining the relative objects using the geometry of triangles. It requires the inter-node distances between the node with unknown location and its neighbors with known locations (at least three neighbors in 2D space and at least four neighbors in 3D space). The minimization problem with the performance function calculated as a difference between the measured and estimated distances is formulated and solved. The concept of iterative multitrilateration is to repeat trilateration for an increased number of anchor nodes (every iteration, each node with an estimated position becomes a new anchor).

In case of schemes using stochastic optimization we formulate the optimization problem with the performance measure J considering estimated Euclidean distances for all pairs of nodes in a given network. As mentioned in the previous section we consider a WSN formed by M anchor nodes with known position expressed as l-dimensional coordinates $a_k \in \mathbf{R}^l$, $k = 1, \ldots, M$ and N non-anchor nodes $x_i \in \mathbf{R}^l$, $i = 1, \ldots, N$ with unknown locations. Our goal is to estimate the coordinates of non-anchor nodes. We can formulate the following minimization problem taking the above assumptions into account:

$$\min_{\hat{x}} \left\{ J = \sum_{k=1}^{M} \sum_{j \in N_k} \left(||a_k - \hat{x}_j||_2 - \tilde{d}_{kj} \right)^2 + \sum_{i=1}^{N} \sum_{j \in N_i} \left(||\hat{x}_i - \hat{x}_j||_2 - \tilde{d}_{ij} \right)^2 \right\}, \quad (4)$$

where $\hat{x}_i$ and $\hat{x}_j$ denote estimated positions of nodes i and j, $\tilde{d}_{kj}$ and $\tilde{d}_{ij}$ distances between pairs of nodes (k, j) and (i, j) calculated based on radio signal measurements, $N_k = \{(k, j) : d_{kj} \leq r\}$, $N_i = \{(i, j) : d_{ij} \leq r\}$ sets of neighbors of anchor and non-anchor nodes $(j = 1 \ldots, N)$, and r maximal transmission range (estimated based on available measurements).

The nonlinear optimization solvers can be used to solve the problem (4). The *PCM* module provides implementations of two optimization algorithms: SA and simple GA. Moreover, we have developed a hybrid solution that uses a combination of the trilateration method, along with stochastic optimization. We have implemented two versions of our scheme: TSA that combines multitrilateration and simulated annealing and TGA that combines multitrilateration and genetic algorithm. Although this hybrid localization system was described in [16], we provide a brief summary of this technique.

Our hybrid localization system operates in two phases:

Phase 1 – the auxiliary solution (localization) is provided using the geometry of triangles.

Phase 2 – the results of initial localization are refined using the stochastic optimization (simulated annealing in TSA and simple genetic algorithm in TGA).

Algorithm 1 Simulated annealing algorithm

1: $T = T_0$, T_0 – initial temperature, T_f – final temperature
2: $\tau = \tau_0$, τ_0 – initial step value
3: **while** $T > T_f$ **do**
4:    **for** $i = 1$ to $P \cdot N$ **do**
5:       select a node to perturb
6:       generate a random direction and move a node at distance equal to step τ
7:       evaluate the change in the cost function, ΔJ
8:       **if** $(\Delta J \leq 0)$ **then**
9:          //downhill move $\Rightarrow$ accept it
10:          accept this perturbation and update the solution
11:       **else**
12:          //uphill move $\Rightarrow$ accept with probability
13:          pick a random probability $rp = \text{uniform}(0,1)$
14:          **if** $(rp \leq exp(-\Delta J/T))$ **then**
15:             accept this perturbation and update the solution
16:          **else**
17:             reject this perturbation and keep the old solution
18:          **end if**
19:       **end if**
20:    **end for**
21:    change the temperature: $T_{new} = \alpha \cdot T$, $T = T_{new}$
22:    change the distance $\tau_{new} = \beta \cdot \tau$, $\tau = \tau_{new}$
23: **end while**

PCM provides an implementation of basic SA algorithm with one modification – the cooling process is slowed down. At each value of the coordinating parameter T (temperature), not one but $P \cdot N$ non-anchor nodes are randomly selected for modification (where P is a reasonably large number to make the system into thermal equilibrium). The general scheme of our implementation – the simulated annealing method – is presented in Algorithm 1. The efficiency and robustness of this algorithm strongly depend on control parameters α, β, τ_0, P, T_0, T_f depicted in Algorithm 1. All these parameters influence the speed of convergence and accuracy of the solution. To obtain the general purpose algorithm their parameters should be tuned up for diverse network topologies.

The implementation of a simple genetic algorithm as described in [7] is provided in the *PCM* module. The abstract representations of candidate solutions (chromosomes) are vectors of coordinates of all non-anchor nodes: $[x_1^1, \ldots, x_1^l, x_2^1, x_2^l, \ldots, x_N^1, \ldots, x_N^l]$, $x_i^j \in \mathbf{R}$ where $i = 1, \ldots, N$, $j = 1, \ldots, l$. The initial population consists of a set of such chromosomes. The fitness function is defined in (4). The tournament selection of size equal to two is used at the reproduction stage. The crossover operator is defined as discrete recombination similar to elements exchanging applied to binary vectors. Both coordinates are recombined simultaneously. The mutation operator modifies the components of a given chromosome by adding a vector of ge-

nerated $l \cdot n$ Gaussian random variables. The elitist succession model is chosen. The TGA algorithm efficiency and robustness strongly depend on a size of population and control parameters of the genetic operators.

3.5 Runtime Platform

The main goal of the runtime platform is to provide an environment for tuning and testing the localization solvers integrated with the WSNLS framework on a parallel computer or a cluster of computers. It is built by two main components: *Distributed Computing Manager* (*DCM*) and one or several *Computational Servers* (*CS*).

3.5.1 Distributed Computing Manager

The *Distributed Computing Manager (DCM)* is responsible for the calculation task splitting into subtasks and allocating the subtasks to processors or computers, managing of calculations and inter-processes communication, and balance the load of all computing nodes. Moreover, *DCM* is responsible for displaying the results of calculations and monitoring the status of all calculated tasks. One *DCM* is capable to manage many computational servers.

3.5.2 Computational Server

The *Computational Server* manages the calculation process concerned with the estimation of the geographic coordinates of nodes in a given network, see Figure 1. *CS* performs the following operations: load initial parameters of the solvers and WSN topology, run the *PCM* module to calculate the locations of nodes and return calculation results to the *DCM* component. Many *CS* can be executed during one simulation, each on different, remote machine. The computations are done by dedicated calculation threads. The number of threads should not exceed the number of processor cores in the computing system. The information about available number of cores together with the information about a communication port and IP address are stored in the XML configuration file. To reduce the interprocesses communication the network topology is sent immediately after the start of calculations. Next, it is recorded at the local *CS* repository. Moreover, each computational server has its own methods repository, so it is possible to add new localization solver by providing new *Computational Server* implementation without modifying the running ones. The architecture of the computational server is depicted in Figure 3.

3.5.3 Communication Protocol

The client-server communication model is implemented in the WSNLS system. The communication between *DCS* and each computational server *CS* is based on TCP/IP and BSD sockets, since we assumed that provided solution should be open architecture and should not be bounded up with any dedicated infrastructure. The
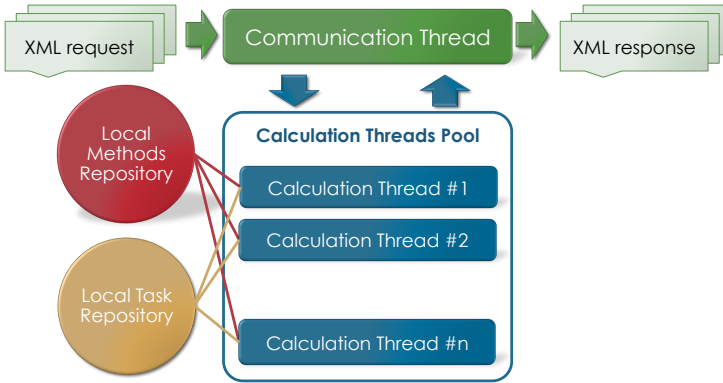
Figure 3. Architecture of the computational server

communication protocol is implemented in the XML scheme. The concept of this protocol was to develop the simple mechanism that fulfills the following requirements:

- flexibility – the protocol should be easy to modify and extend with new messages,
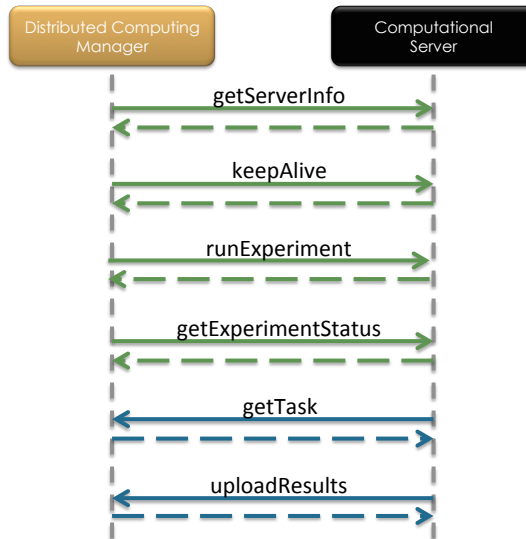- failure resistance – the protocol should be robust as much as possible.

Figure 4. Messages defined in our communication protocol

Our communication protocol is formed by the following messages (Figure 4):

getServerInfo [DCM ⇒ CS] question about server configuration, such as number
of cores etc,

keepAlive [DCM ⇒ CS] link checking,

runExperiment [DCM ⇒ CS] order to run computations specifying task, method,
its parameters and number of runs,

getExperimentStatus [DCM ⇒ CS] question about computation progress,

getTask [CS ⇒ DCM] order to download task from *DCM*,

uploadResults [CS ⇒ DCM] order to upload results to *DCM*.

4 NUMERICAL RESULTS

To evaluate the efficiency of the WSNLS framework in wireless sensor networks
localization, and to show the possible range of applications two series of experiments
were performed. All tests were carried out on our computer cluster system that
integrates two types of CPUs: 12 servers with Intel Xeon X5650 and 12 servers with
AMD Opteron 6172 processors.

4.1 Localization System Tuning

The goal of the first series of experiments was to show how the values of parameters
specific to the localization solvers influence the accuracy of the estimated location
of a node. To evaluate the performance of tested localization systems, we used the
mean error between the estimated and the true location of the non-anchor nodes in
the network, defined as follows:

$$LE = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{(||\hat{x}_i - x_i||)^2}{r_i^2} \cdot 100\,\%, \tag{5}$$

where N denotes the number of non-anchor nodes in a network, LE denotes a loca-
lization error, x_i the true position of the node i in the network, $\hat{x}_i$ estimated location
of the node i (solution of the localization system) and r_i the radio transmission range
of the node i. The localization error LE is expressed as a percentage error. It is
normalized with respect to the radio range to allow comparison of results obtained
for different size and range networks.

Three network system configurations with different size of the network (number
of nodes), number of anchor nodes and anchor nodes deployment were considered.
The TSA algorithm was used to calculate the coordinates of nodes forming the tested
networks. Four control parameters, i.e., *alpha*, *beta*, *tau*, and *temperature* (T_f) spe-
cific to our implementation of the simulated annealing algorithm (see Algorithm 1)
were considered. The number of experiments was equal to 48 000, each for different
set of parameters. The ranges of tested parameters values are presented in Table 1.

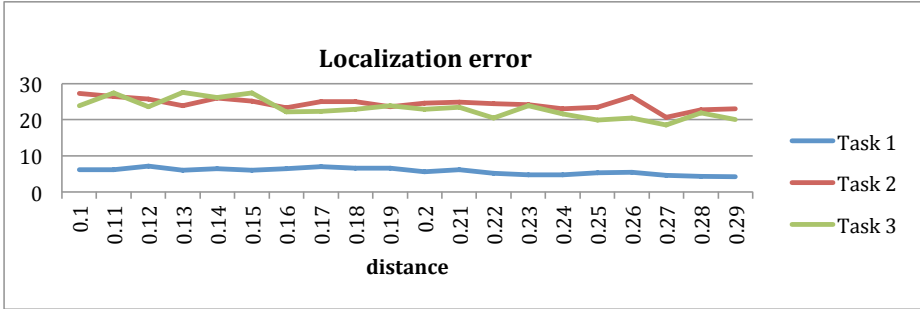| Parameter	Begin	End	Step	Multiply
alpha	0.6	0.98	0.02	
beta	0.6	0.98	0.02	
final temperature	1e−9	1e−14		0.1
tau	0.1	0.29	0.01	

Table 1. Parameters range in batch test

Figure 5. Localization error as a function of *alpha* parameter

Figures 5 and 6 depict the localization error as a function of *alpha* and *tau* parameters (values of *beta* and *temperature* were constant). The presented calculation results demonstrate that localization system based on the simulated annealing algorithm is very sensitive to the control parameter *alpha*. The closest to accurate location estimates were achieved for *alpha* = 0.96. In this experiment *beta* was equal to 0.98. The pair of values of parameters *alpha* = 0.96 and *beta* = 0.98 formed the optimal set of values which allows to obtain the smallest localization error for considered tasks. This result is in accordance with intuition – it is safer to change the temperature in the SA algorithm slowly, although in some cases it is possible to obtain better accuracy for different values of *alpha*.
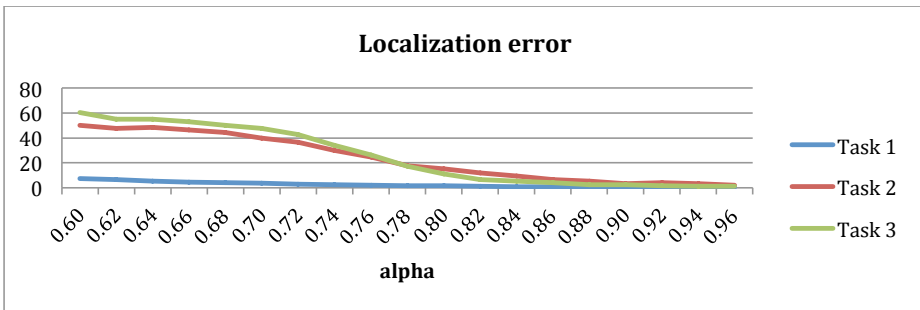

Figure 6. Localization error as a function of *tau* parameter

Moreover, the tests confirmed that better results can be obtained when the step value (*tau*) of node movement is shrunk slower than the coordinating parameter (*temperature*) – that is *beta > alpha*. The impact of *tau* parameter is definitely less significant than *alpha* and *beta* parameters, at least for *alpha* and *beta* values exceeding 0.9 (see Figure 6).

4.2 Localization Systems Comparison

The aim of the second series of experiments was to compare the accuracy and performance of localization systems based on stochastic optimization with simple trilateration method. Four methods were considered – the trilateration scheme, the pure simulated annealing SA and hybrid techniques TSA and TGA. The simulations were performed for WSNs consisting of 200–1 000 nodes with randomly generated positions (both anchor and non-anchor nodes) in a square region $[0, 1] \times [0, 1]$. Basic characteristics of test networks are collected in Table 2. The connectivity measure in Table 2 denotes the average number of connections between nodes in a given network.

Network	Task200	Task500	Task1000
Number of nodes	200	500	1 000
Number of anchors	20	50	100
Connectivity measure	17.57	19.73	18.70

Table 2. Basic characteristics of test networks

Selected results, i.e., averaged values of localization errors computed due to (5) for all tested localization systems, various network topologies and five executions of all tests are collected in Table 3 and Figure 7. The results of calculations show that all localization systems based on stochastic optimization produce much more accurate localization with respect to the simple trilateration. A localization accuracy strongly depends on the applied optimization solver - the SA algorithm gives better solution than GA. The most promising approach is the hybrid scheme combining the trilateration and the simulated annealing method. Regardless of network size for the TSA scheme the localization error does not exceed 0.2 %. It is worth to mention that such good result can be obtained for optimal values of all control parameters of the SA solver. The extensive tuning procedure was applied to TSA method before the localization step.

Table 4 and Figure 8 demonstrate the computation times in case of all localization systems and various network topologies. As expected the calculation time for the trilateration scheme is almost unnoticeable as it does not exceed one second for every task from the testing set. The localization systems utilizing stochastic optimization require more extensive calculations and are thus much more time consuming. Unfortunately, the computation time strongly depends on the dimension of the tested network and increases faster then linearly with the number of nodes.

Network	Task200	Task500	Task1000
Trilateration	13.80	14.05	12.45
SA	3.38	2.40	0.60
TSA	0.14	0.04	0.19
TGA	1.85	5.29	5.03

Table 3. Localization error – LE (5)

Figure 7. Localization error for three test networks and four localization systems

The TSA method is the fastest one but the differences between computation times of SA, TSA and TGA decrease for higher dimension networks.

Network	Task200	Task500	Task1000
Trilateration	0.02	0.03	0.12
SA	7.20	27.49	49.94
TSA	0.57	3.95	38.61
TGA	3.09	14.30	50.00

Table 4. Calculation time (in seconds)

5 CONCLUSIONS AND FUTURE WORKS

We have presented the design and evaluation of our WSNLS parallel framework for localization of nodes that form a wireless sensor network. Our tool can support the design and development of novel localization systems and can be used to select the best localization method for a given WSN application. WSNLS can be easily extended with localization systems developed by other researchers after their adaptation to our software environment.

The presented results of numerical experiments indicate that metaheuristics are versatile and attractive techniques to WSN localization. However, it is commonly known that efficiency and robustness of these methods strongly depend on different control parameters of the algorithm. To obtain the general purpose algorithm to
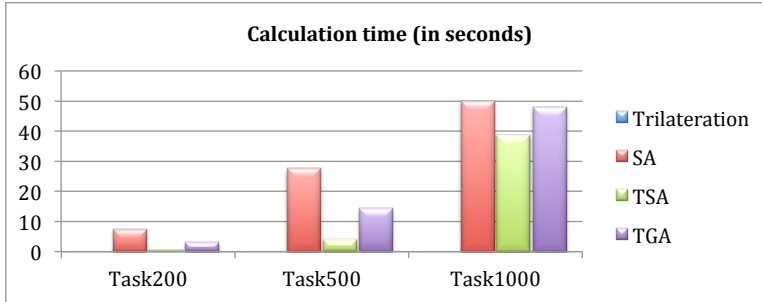
Figure 8. Calculation time for three networks with different number of nodes

solve the localization problem the parameters should be tuned for various network size and topology. Unfortunately, it is a time consuming process. The parallel implementation reduces the calculation time and improves calculation robustness. As presented in this paper our WSNLS parallel framework can successfully support the calculation of the optimal values of control parameters of available solvers before their usage in a real network.

Acknowledgment

REFERENCES

[1] AKYILDIZ, I. F.—VURAN, M. C.: Wireless Sensor Networks. John Wiley & Sons, Ltd., West Sussex, UK 2010.

[2] AWAD, A.—FRUNZKE, T.—DRESSLER, F.: Adaptive Distance Estimation and Localization in WSN Using RSSI Measures. In 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools 2007, pp. 471–78.

[3] BACCOUR, N.—KOUBAA, A.—MOTTOLA, L.—ZUNIGA, M.—YOUSSEF, H.—BOANO, C. A.—ALVES, M.: Radio Link Quality Estimation in Wireless Sensor Networks: A Survey. ACM Transactions on Sensor Networks, Vol. 8, 2012, No. 4.

[4] BARSOCCHI, P.—LENZI, S.—CHESSA, S.—GIUNTA, G.: Virtual Calibration for RSSI-Based Indoor Localization with IEEE 802.15.4. In ICC '09 – IEEE International Conference on Communications, pp. 1–5.

[5] BENKIC, K.—MALAJNER, M.—PLANINSIC, P.—CUCEJ, Z.: Using RSSI Value for Distance Estimation in Wireless Sensor Networks Based on ZIGBEE. In 15th International Conference on Systems, Signals and Image Processing 2008, pp. 303–306.

[6] BISWAS, P.—YE, Y.: Semi-Definite Programming for Ad hoc Wireless Sensor Network Localization. In Proc. of IPSN '04: Information processing in sensor networks 2004, pp. 46–54.

[7] GOLDBERG, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Studies in Applied Mathematics, Addison-Wesley 1989.

[8] KANNAN, A. A.—MAO, G.—VUCETIC, B.: Simulated Annealing Based Localization in Wireless Sensor Network. In LCN '05: Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary, 2005, pp. 513–514.

[9] KANNAN, A. A.—MAO, G.—VUCETIC, B.: Simulated Annealing Based Wireless Sensor Network Localization with Flip Ambiguity Mitigation. In 63rd IEEE Vehicular Technology Conference 2006, pp. 1022–1026.

[10] KARL, H.—WILLIG, A.: Protocols and Architectures for Wireless Sensor Networks. Wiley 2005.

[11] MAO, G.—FIDAN, B.: Localization Algorithms and Strategies for Wireless Sensor Networks. Information Science Reference, Hershey, USA 2009.

[12] MAO, G.—FIDAN, B.—ANDERSON, B. D. O.: Wireless Sensor Network Localization Techniques. Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 51, 2007, No. 10, pp. 2529–2553.

[13] MARKS, M.—NIEWIADOMSKA-SZYNKIEWICZ, E.: Self-Adaptive Localization Using Signal Strength Measurements. In SENSORCOMM 2011 – the Fifth International Conference on Sensor Technologies and Applications, pp. 73–78.

[14] NICULESCU, D.—NATH, B.: Ad Hoc Positioning System (APS). In GLOBECOM: Global Telecommunications Conference 2001, Vol. 5, pp. 2926–2931.

[15] NIEWIADOMSKA-SZYNKIEWICZ, E.: Localization in Wireless Sensor Networks: Classification and Evaluation of Techniques. International Journal of Applied Mathematics and Computer Science, Vol. 22, 2012, No. 2, pp. 281–297.

[16] NIEWIADOMSKA-SZYNKIEWICZ, E.—MARKS, M.: Optimization Schemes for Wireless Sensor Network Localization. International Journal of Applied Mathematics and Computer Science, Vol. 19, 2009, pp. 2, pp. 291–302.

[17] NIEWIADOMSKA-SZYNKIEWICZ, E.—MARKS, M.—KAMOLA, M.: Localization in Wireless Sensor Networks Using Heuristic Optimization Techniques. Journal of Telecommunications and Information Technology, Vol. 4, 2011, pp. 55–64.

[18] RAMADURAI, V.—SICHITIU, M. L.: Localization in Wireless Sensor Networks: A Probabilistic Approach. In Proceedings of International Conference on Wireless Networks (ICWN 2003), pp. 300–305.

[19] RAPPAPPORT, T. S.: Wireless Communications: Principles and Practice. Communications Engineering and Emerging Technologies Series, Prentice Hall, 2nd edition 2002.

[20] SRINIVASAN, K.—LEVIS, P.: RSSI Is Under Appreciated. In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets), Cambridge, USA, 2006.

[21] ZHANG, X.—WU, Y.—WEI, X.: Localization Algorithms in Wireless Sensor Networks Using Nonmetric Multidimensional Scaling with RSSI for Precision Agriculture. In The 2nd International Conference on Computer and Automation Engineering (ICCAE) 2010, Vol. 5, pp. 556–559.

[22] ZUNIGA, M.—KRISHNAMACHARI, B.: Analyzing the Transitional Region in Low Power Wireless Links. In First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON) 2004, pp. 517–526.

Michał Marks received M. Sc. (2007) in computer science from the Warsaw University of Technology. He is Ph. D. student at the Institute of Control and Computation Engineering, the Warsaw University of Technology. Since 2007 he is with Research and Academic Computer Network (NASK). He is the author and co-author of over 20 journal and conference papers. His research area focuses on wireless sensor networks, global optimization, distributed computation in CPU and GPU clusters, decision support and machine learning.

Ewa Niewiadomska-Szynkiewicz received her Dr. Sc. in 2005, and her Ph. D. in 1995. She is Professor of control and information engineering at Warsaw University of Technology, and Head of the Complex Systems Group. She is also the Director for Research of Research and Academic Computer Network (NASK). She is the author and co-author of three books and over 120 journal and conference papers. Her research interests focus on complex systems modeling and control, computer simulation, global optimization, parallel computation and computer networks. She was involved in a number of research projects including EU projects, coordinated the Groups activities, managed organization of a number of national-level and international conferences.

Joanna Kołodziej graduated in mathematics from Jagiellonian University in Cracow in 1992, where she also obtained her Ph. D. in Computer Science in 2004. She joined the Department of Mathematics and Computer Science of the University of Bielsko-Biala as an Assistant Professor in 1997. She has been serving as PC Co-Chair, General Co-Chair and IPC member of several international conferences and workshops including PPSN 2010, ECMS 2011, CISIS 2011, 3PGCIC 2011, CISSE 2006, CEC 2008, IACS 2008–2009, ICAART 2009-2010. She is the Managing Editor of IJSSC Journal and serves as an EB member and guest editor of several peer-reviewed international journals.