ENHANCING WSN LOCALIZATION ROBUSTNESS UTILIZING HPC ENVIRONMENT

Michal Marks
Research and Academic Computer Network (NASK)
Wawozowa 18, 02-796 Warsaw, Poland
and
Institute of Control and Computation Engineering,
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
Email: mmarks@elka.pw.edu.pl

KEYWORDS

Wireless Sensor Network, positioning, stochastic optimization, simulated annealing, localization system, distributed computing, HPC.

ABSTRACT

The paper treats the problem of localization in Wireless Sensor Network (WSN). In our work, we present and evaluate *Wireless Sensor Network Localization System*, which offers network models generator along with different localization methods including Trilateration & Simulated Annealing algorithm. The paper describes extension of *WSN Localization System* with modules supporting distributed computing in HPC environment. A provided case study concentrate on improving algorithms robustness through parallel solving a huge number of localization tasks. *WSN Localization System* in distributed version is used for generation a set of test networks with various topology parameters and solving the created localization tasks with very different values of method parameters. Applying distributed computing in our HPC infrastructure allows to speedup calculations by two orders of magnitude.

INTRODUCTION TO WSN LOCALIZATION

The goal of localization is to assign geographic coordinates to each node in the sensor network in the deployment area. Wireless sensor network localization is a complex problem that can be solved in different ways, Karl and Willig (2005). A number of research and commercial location systems for WSNs have been developed. They differ in their assumptions about the network configuration, distribution of calculation processes, mobility and finally the hardware's capabilities, Mao et al. (2007); Awad et al. (2007); Zhang et al. (2010).

Recently proposed localization techniques consist in identification of approximate location of nodes based on merely partial information on the location of the set of nodes in a sensor network. An anchor is defined as a node that is aware of its own location, either through GPS or manual pre-programming during deployment. Identification of the location of other nodes is up to an algorithm locating non-anchors. Considering hardware's capabilities of network nodes we can distinguish two classes of methods: range based (distance-based) methods and range free (connectivity based) methods.

The former is defined by protocols that use absolute point to point distance estimates (ranges) or angle estimates in location calculation. The latter makes no assumption about the availability or validity of such information, and use only connectivity information to locate the entire sensor network. The popular range free solutions are hop-counting techniques. Distance-based methods require the additional equipment but through that much better resolution can be reached than in case of connectivity based ones. In our works we concentrate on range based methods.

The paper is structured as follows: at the beginning we formulate the distance-based localization problem. Next, we provide a short overview of our software environment for WSN localization and an extension applied to our software in order to utilize HPC environment. Finally, we provide a case study results and conclusions.

DISTANCE-BASED LOCALIZATION PROCESS

Distance-based localization is a complex problem and solving it requires to combine two techniques: signal processing and algorithms transforming measurements into the coordinates of the nodes in the network. Hence, distance-based localization schemes operate in two stages.

- *Distance estimation stage* – estimation of inter-node distances based on inter-node transmissions.
- *Position calculation stage* – calculation of geographic coordinates of nodes forming the network.

Distance estimation stage

As it was mentioned in the introduction using range based methods we can reach much better resolution than in case of range free ones. However in order to do that the additional equipment is usually required. Each of popular techniques – widely described in literature, Karl and Willig (2005); Mao et al. (2007), such as Angle of Arrival

Proceedings 26th European Conference on Modelling and Simulation ©ECMS Klaus G. Troitzsch, Michael Möhring, Ulf Lotzmann (Editors)
ISBN: 978-0-9564944-4-3 / ISBN: 978-0-9564944-5-0 (CD)

(AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA) needs an additional stuff such as antennas or accurately synchronized clocks. The only exception from these requirements is a Received Signal Strength Indicator (RSSI) technique.

RSSI is considered as the simplest and cheapest method amongst the wireless distance estimation techniques, since it does not require additional hardware for distance measurements and is unlikely to significantly impact local power consumption, sensor size and thus cost. Main disadvantage of using RSSI is low accuracy. In respect to wireless channel models provided in literature, Rappapport (2002), received power should be a function of distance. However, the RSSI values have a high variability and they cannot be treated as a good distance estimates, Ramadurai and Sichitiu (2003); Benkic et al. (2008). Nevertheless some authors indicate that new radio transceivers can give RSSI measurements good enough to be a reasonable link estimator, Srinivasan and Levis (2006); Barsocchi et al. (2009).

The signal propagation model outlined in Rappapport (2002); Marks and Niewiadomska-Szynkiewicz (2011) allows to estimate the distance if the strength of received signal is known. The objective of the distance estimation stage is to tune up the parameters of propagation model. It should be underline that the aim of this procedure is obtaining the smallest errors in distances estimations as it is impossible to achieve accurate results due to RSSI inaccuracy. In our previous paper, Marks and Niewiadomska-Szynkiewicz (2011), we provided three methods of distance estimation wrt a given network topology and deployment area – Ordinary Least Square Method (OLS), Weighted Least Square Method (WLS) and Geometric Combined Least Square Method (GCLS).

Position calculation stage

In the position calculation stage the measurements of inter-node distances are used to estimate the coordinates of non-anchor nodes in the network. Let us consider a WSN formed by M sensors (anchor nodes) with known position expressed as l-dimensional coordinates $a_k \in \mathbf{R}^l,\ k = 1, \ldots, M$ and N sensors (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 optimization problem with the performance measure J considering estimated Euclidean distances of all neighbor nodes:

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

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 (assessed based on available measurements).

The stochastic optimization algorithms can be used to solve the problem (1). Kannan et al. (2005) present the results of location calculation for simulated annealing method. We propose the hybrid technique that uses a combination of the trilateration method, along with simulated annealing (TSA: Trilateration & Simulated Annealing). TSA was described in details in Marks and Niewiadomska-Szynkiewicz (2007). It operates in two phases:

- *Phase 1* – the auxiliary solution (localization) is provided using the geometry of triangles.
- *Phase 2* – the solution of the phase 1 is improved by applying stochastic optimization.

From the perspective of algorithm robustness especially the second phase is important. It is based on simulated annealing (SA) which is a well known heuristic used to solve the localization problem (1), Kannan et al. (2005); Mao et al. (2007). It is implemented as a computer simulation of a stochastic process. It performs point-to-point transformation. Our implementation of SA algorithm is a classical version of SA 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 N denotes the number of sensors with unknown positions in the network and P is a reasonably large number to make the system into thermal equilibrium). The general scheme of SA algorithm is presented in Algorithm 1.

TSA algorithm efficiency and robustness strongly depend on control parameters $\alpha, \beta, \Delta d_0, P, T_0, T_f$ specific to the simulated annealing algorithm used in the second phase of TSA, and depicted in Algorithm 1. All these

Algorithm 1 Simulated annealing algorithm

```
T = T_0, T_0 – initial temperature, T_f – final temperature
Δd = Δd_0, Δd_0 – initial move distance
while T > T_f do
    for i = 1 to P · N do
        select a node to perturb
        generate a random direction and move a node at distance Δd
        evaluate the change in the cost function, ΔJ
        if (ΔJ ≤ 0) then
            //downhill move ⇒ accept it
            accept this perturbation and update the solution
        else
            //uphill move ⇒ accept with probability
            pick a random probability rp = uniform(0,1)
            if (rp ≤ exp(−ΔJ/T)) then
                accept this perturbation and update the solution
            else
                reject this perturbation and keep the old solution
            end if
        end if
    end for
    change the temperature: T_new = α · T, T = T_new
    change the distance Δd_new = β · Δd, Δd = Δd_new
end while
```

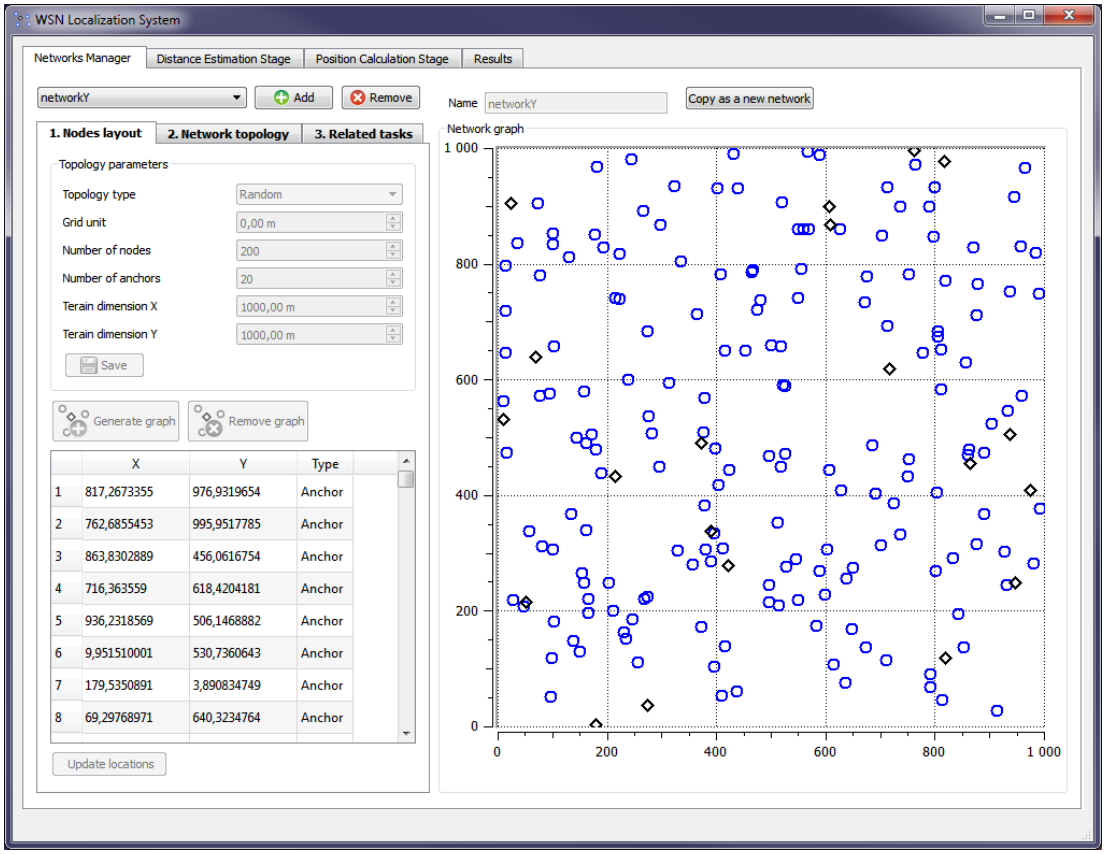

Figure 1: Networks manager in WSN Localization System

parameters influence the speed of convergence and accuracy of the solution. To obtain the general purpose algorithm the values of them should be tuned up for diverse network topologies.

SIMULATION ENVIRONMENT

In order to evaluate our two-phase method new software tool – *WSN Localization System* was created. *WSN Localization System* supports not only the both phases of localization process but it offers also the network models generator. The system architecture is depicted in Figure 2. A user-friendly graphical interface (GUI) for interacting with our system is given. Except the GUI and the database, used for storing all data connected with networks, tasks and localization results, localization system is composed of three main components: *Networks Manager*, *Distance Estimation Module* and *Position Calculation Module*.

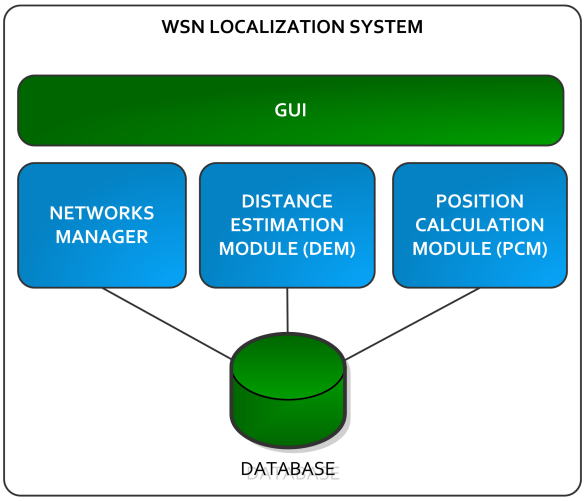

Figure 2: WSN Localization System components

Networks Manager

Networks Manager (Fig. 1) 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 the proper modeling of low-power links is very difficult since the links characterization depends on radio chips (e.g., TR1000, CC1000, CC2420, etc), operational environments (indoor, outdoor) and many other parameters such as traffic load or radio channel – Baccour et al. (2012). In our software we decided to provide models based on *Link Layer Model for MATLAB* provided by Zuniga and Krishnamachari (2004). *Networks Manager* focus on wireless channel modeling and no radio modulation and encoding are considered. In the future *Networks Manager* will be extended to provide data gathering from real-life deployments.

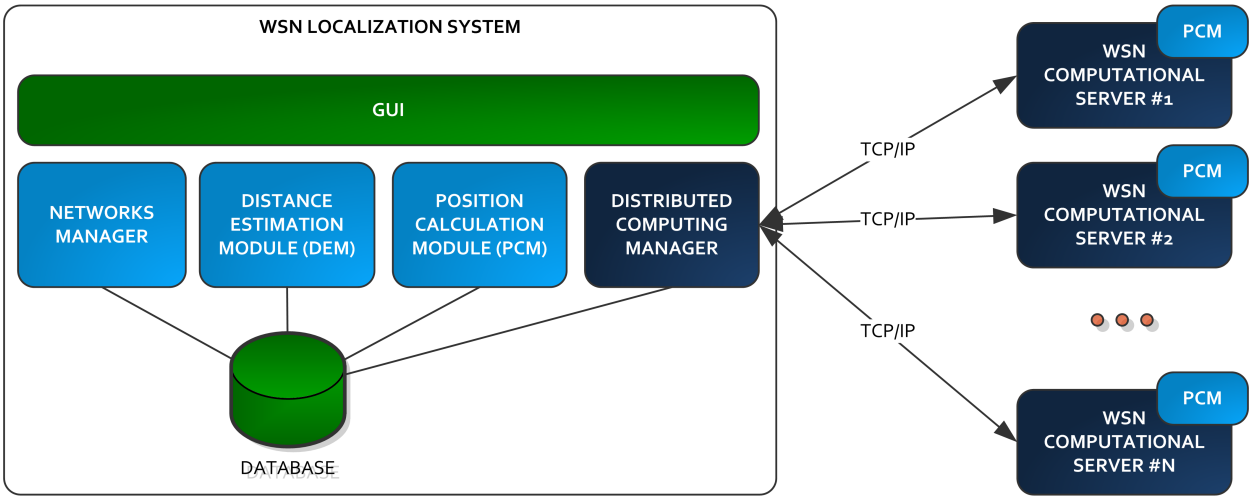

Figure 3: Distributed system architecture

Distance Estimation Module

Distance Estimation Module (DEM) provides optimization methods transforming RSSI measurements into internode distances estimations. At present *DEM* has registered three approaches to distance estimation: Ordinary Least Square Method (OLS), Weighted Least Square Method (WLS) and Geometric Combined Least Square Method (GCLS). More information about distance estimation stage can be found in Marks and Niewiadomska-Szynkiewicz (2011).

Position Calculation Module

Position Calculation Module (PCM) is the main component of our environment as it provides methods for estimating the coordinates of non-anchor nodes in the network using inter-node distances. PCM is realized in the object-oriented way and it can be easily extended with new localization algorithms. Currently TSA (Trilateration & Simulated Annealing) and SA (Simulated Annealing) methods are supported, in the near future TGA (Trilateration & Genetic Algorithm) method will be added. More information about position calculation methods can be found in Niewiadomska-Szynkiewicz and Marks (2009).

DISTRIBUTED COMPUTING EXTENSION

Localization accuracy strongly depends on the measurement errors, network density and anchor nodes location. In paper Niewiadomska-Szynkiewicz and Marks (2009) we provided an extensive analysis of impact of anchor nodes distribution on localization accuracy. However achieving high quality results for very different network topologies in many cases require running localization methods with different values of parameters. For example TSA method can be tuned up by setting up almost a dozen parameters such as: cooling scheme, initial temperature, final temperature, shrinking factors for temperature and movement distances.

On the other hand localization method should guarantee achieving reasonable results for different networks without any tuning, as it is often impossible to select appropriate values in the unknown environment. In order to provide a set of "safety settings" we decided to generate a set of test networks with various topology parameters and to solve the created test tasks with different values of method's parameters. This experiment allows us for improving algorithms robustness but requires solving millions of tasks. Of course this can be done on single machine but it can take a few days to solve all the tasks. Therefore we decided to extend our *WSN Localization System* to include new capabilities connected with distributed computing.

System Architecture

The new capabilities required preparation a new system architecture with *Distributed Computing Manager* module inside simulator and *WSN Computational Server* application. We decided to use client-server communication model as in our system the computational task can be easily decomposed into the set of independent subtasks. The system architecture is presented in Figure 3.

The communication between System and computational server is based on TCP/IP, since we assumed that provided solution shouldn't be bounded up with any special infrastructure such as InfiniBand or protocol like MPI. Of course one System is capable to cooperate with many computational servers.

Distributed Computing Manager

Distributed Computing Manager is the new component in *WSN Localization System* that is responsible for client-server communication and calculation management. The main functionalities of the module are: tasks splitting, displaying results of calculations and presenting current status of each task on computational server. The *Distributed Computing Manager* component manages execution of all tasks assigned to distributed processing,

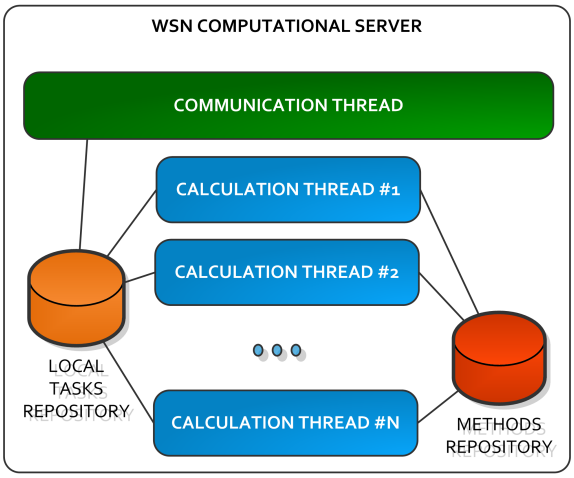

Figure 4: Computational Server Architecture

both in single and batch mode. Moreover, Manager is responsible for computational resources management in order not to overload WSN Computational Servers.

WSN Computational Server

WSN Computational Server is a new application which can be run on remote machines. The role of *WSN Computational Server* is to receive calculation request, realize the experiment and return response to System. The program doesn't have any interface. The computations are done by dedicated calculation threads. The number of threads shouldn't exceed the number of processor cores – the information about available number of cores is stored in XML configuration file. The same configuration file stores also information about port and IP address the communication thread should operate. Each computational server has its own local tasks repository where the network topologies are stored in order to reduce communication – usually only task identifier is sent and there is no need to transfer the whole task descriptor. The task data are transmitted only during running first experiment with appropriate task. It allows not only to reduce communication but also provides a way for new task distribution for running *WSN Computational Servers*. Each computation server has also its own methods repository, so it is possible to add new localization method by providing new *WSN Computational Server* implementation without modifying the running ones. The architecture of computational server is depicted in Figure 4.

Communication Protocol

The communication is done in a master-slave scheme. It is the natural protocol for applications with data decomposition into blocks and iterative calculations. An XML-based communication protocol is proposed to perform communication between System and computational servers. It is based on the TCP/IP protocol and BSD sockets. Our goal was to apply 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.

The following messages are supported by communication protocol (see Figure 5):

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

keepAlive [DCM ⇒ server] link checking,

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

getExperimentStatus [DCM ⇒ server] question about computation progress,

getTask [server ⇒ DCM] order to download task from System,

uploadResults [server ⇒ DCM] order to upload results to System.

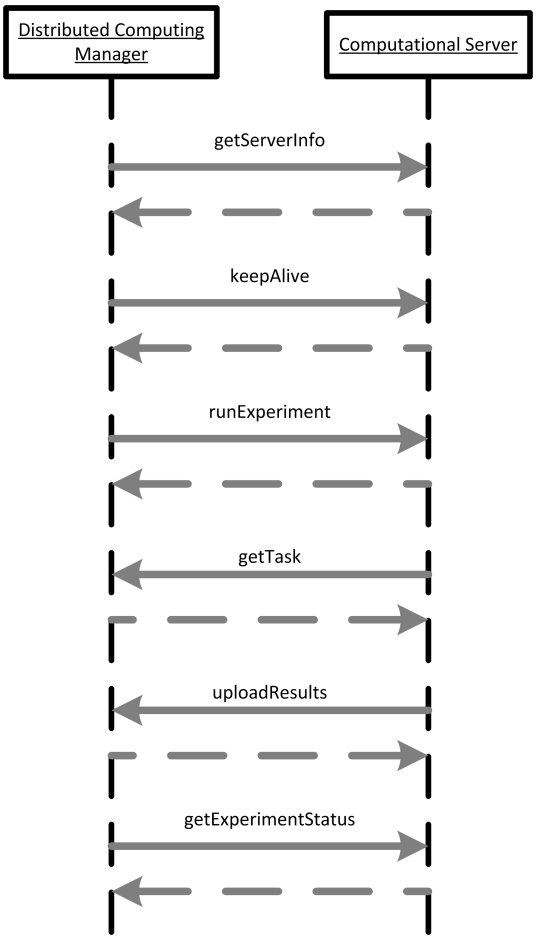

Figure 5: Messages defined in communication protocol.

An example of runExperiment message is presented in Figure 6.

```
<message messageId="00000001">
  <request type="runExperiment">
    <task id="evenly.ols" />
    <method type="TSA">
      <param name="fti" value="4" />
      <param name="alpha" value="0.94" />
      <param name="beta"  value="0.98" />
      <param name="initial_temp" value="0.1" />
    </method>
    <experiment runs="5" />
  </request>
</message>
```

Figure 6: An example of XML message.

NUMERICAL RESULTS

To demonstrate possibilities of our software three tasks connected with three different network topologies were solved with 48000 different values of parameters. The range of tested parameter values is presented in Table 1.

Table 1: Parameters range in batch test

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
distance	0.1	0.29	0.01	

Figures 7 and 8 depict the localization error as a function of alpha and distance parameter. For both charts the values of the rest parameters are constant. In the figure 7 the best solution can be achieved when $alpha = 0.96$. In this experiment $beta$ is equal 0.98. This pair of values forms 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 more safety to change the temperature in SA algorithm slowly, although in some cases it is possible to obtain better accuracy for different values of $alpha$.

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

CONCLUSIONS AND FUTURE WORKS

We have presented the design and evaluation of our *WSN Localization System* extended with distributed computing feature. The software can be used for creation and solving different WSN localization problems using our TSA or SA methods. The software can be easily extended with another methods utilizing the same software framework. Emphasis was placed on the distributed computation modules which allows us for maximizing the methods robustness for different tasks. In our future research, we would like to add additional methods to our software and improve analytical capabilities – displaying charts with best, worst and average solutions.

ACKNOWLEDGMENT

This work was partially supported by Ministry of Science and Higher Education under grant NN514 672940 and the National Centre for Research and Development (NCBiR) under grant O R00 0091 11.

REFERENCES

Awad, A., Frunzke, T., and Dressler, F. (2007). Adaptive distance estimation and localization in wsn using rssi measures. In *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pages 471 –478.

Baccour, N., Koubaa, A., Mottola, L., Zuniga, M., Youssef, H., Boano, C. A., and Alves, M. (2012). Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks*, to appear.

Barsocchi, P., Lenzi, S., Chessa, S., and Giunta, G. (2009). Virtual calibration for rssi-based indoor localization with ieee 802.15.4. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1 –5.

Benkic, K., Malajner, M., Planinsic, P., and Cucej, Z. (2008). Using rssi value for distance estimation in wireless sensor networks based on zigbee. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 303 –306.

Kannan, A. A., Mao, G., and Vucetic, B. (2005). Simulated annealing based localization in wireless sensor network. In *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, pages 513–514, Washington, DC, USA. IEEE Computer Society.

Karl, H. and Willig, A. (2005). *Protocols and Architectures for Wireless Sensor Networks*. Wiley.

Mao, G., Fidan, B., and Anderson, B. D. O. (2007). Wireless sensor network localization techniques. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(10):2529–2553.

Marks, M. and Niewiadomska-Szynkiewicz, E. (2007). Two-phase stochastic optimization to sensor network localization. In *SENSORCOMM 2007: Proceedings of the international conference on Sensor Technologies and Applications*, pages 134–139. IEEE Computer Society.

Marks, M. and Niewiadomska-Szynkiewicz, E. (2011). Self-adaptive localization using signal strength measurements. In *SENSORCOMM 2011, the Fifth International Conference on Sensor Technologies and Applications*, pages 73–78, Nice. IARIA.

Niewiadomska-Szynkiewicz, E. and Marks, M. (2009). Optimization schemes for wireless sensor network localization. *International Journal of Applied Mathematics and Computer Science*, 19(2).

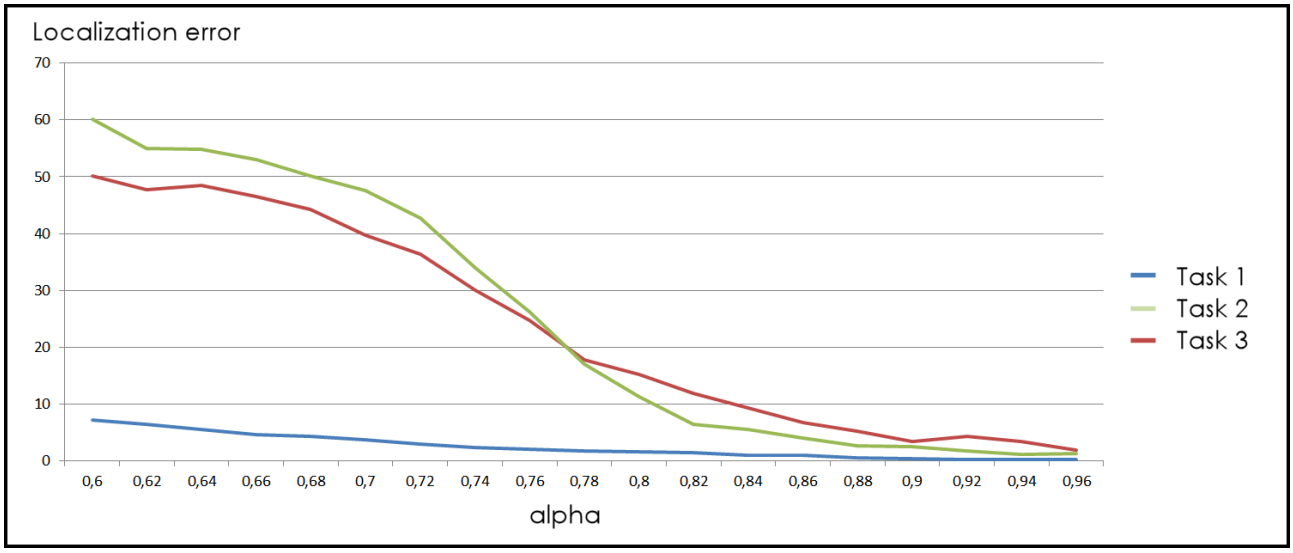

Figure 7: Localization error as a function of alpha parameter.

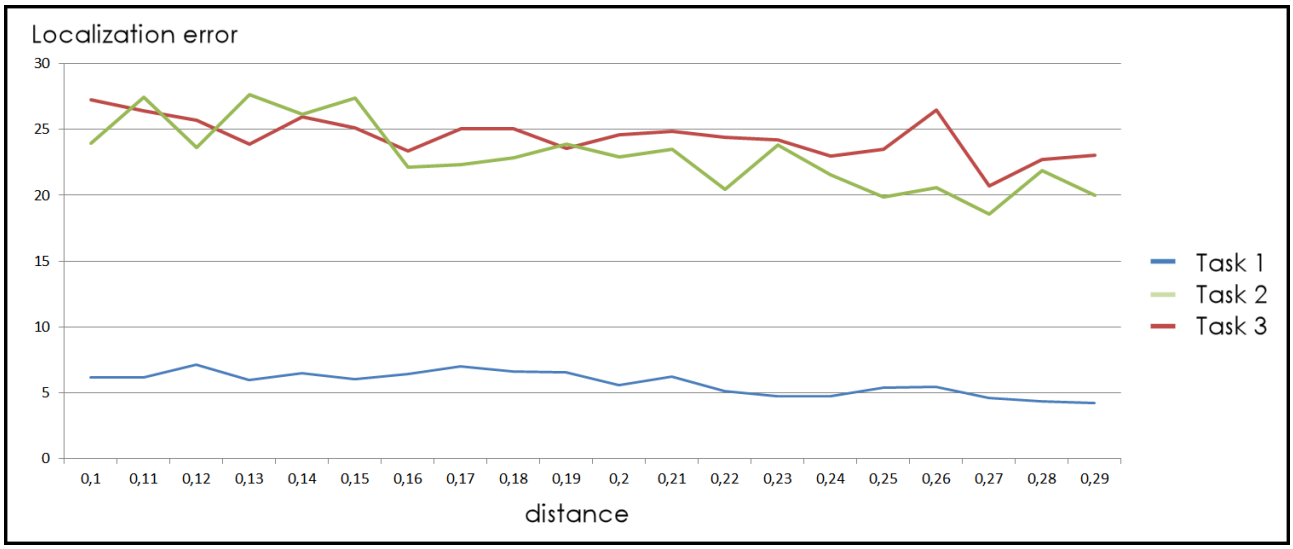

Figure 8: Localization error as a function of distance parameter.

Ramadurai, V. and Sichitiu, M. L. (2003). Localization in wireless sensor networks: A probabilistic approach. In *Proceedings of International Conference on Wireless Networks (ICWN 2003)*, pages 300–305, Las Vegas.

Rappapport, T. (2002). *Wireless communications: principles and practice*. Communications Engineering and Emerging Technologies Series. Prentice Hall, second edition edition.

Srinivasan, K. and Levis, P. (2006). Rssi is under appreciated. In *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets*.

Zhang, X., Wu, Y., and Wei, X. (2010). Localization algorithms in wireless sensor networks using nonmetric multidimensional scaling with rssi for precision agriculture. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, volume 5, pages 556 –559.

Zuniga, M. and Krishnamachari, B. (2004). Analyzing the transitional region in low power wireless links. In *In First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON*, pages 517–526, Santa Clara.

AUTHOR BIOGRAPHIES

MICHAŁ MARKS received his M.Sc. in computer science from the Warsaw University of Technology, Poland, in 2007. Currently he is a Ph.D. student in the Institute of Control and Computation Engineering at the Warsaw University of Technology. Since 2007 with Research and Academic Computer Network (NASK). His research area focuses on wireless sensor networks, global optimization, distributed computation in CPU and GPU clusters, decision support and machine learning. His e-mail is `mmarks@ia.pw.edu.pl`