Bulletin of the *Transilvania* University of Braşov
Series III: Mathematics and Computer Science, Vol. 3(65), No. 1 - 2023, 211-228
https://doi.org/10.31926/but.mif.2023.3.65.1.16

DEVELOPMENT OF A WIRELESS SENSOR NETWORK SIMULATOR USING ENERGY-EFFICIENT TRAFFIC ROUTING METHODS

Ivan S. GAVRILOV[*1]

Abstract

This document represents possible algorithms and methods for the analysis, optimization, and control of wireless sensor networks. The main goal of the work is to find the optimal implementation of the particle swarm optimization method for solving the problem of improving the energy consumption of such networks. The result of the work is a software package for simulating WSN (Wireless sensor network) networks using all the methods of routing and optimization of WSN networks presented in the work. Usage of a combination of routing schemes and different systems of device state management raises the wireless sensor network lifetime and reduces the power consumption of the network. The developed and presented in work network simulator allows any developer to design and test different network topologies with various network parameters. Such software will make it easy to design such networks both for large enterprises and independent developers.

2000 *Mathematics Subject Classification:* 68M18.
Key words: technologies, networks, IoT, optimization, routing, sensors.

1 Introduction

Every year, with the development of technological progress, people are finding more and more ways to optimize and automate their everyday lives. Based on microprocessors, many companies and enthusiasts create small smart devices, which exchange data within the same network or on the internet. These can be various household appliances or sensor devices that can receive different information about the environment and share to other devices or services.

The suboptimal use of resources and production opportunities, the environmental pollution, the lack of monitoring of the state of facilities at factories -

[1*] *Corresponding author*, National Research University "Higher School of Economics", Russia, e-mail: isgavrilov@edu.hse.ru

these problems of modern industry stimulate companies to use more modern informational technologies systems, such as artificial intelligence, robotics, and the internet of things (IoT). To state control of a company's facilities (equipment, conveyors, assembly machines, reactors), sets of wireless or wired sensors, which are connected to each other and information systems which are used to control data from sensors and interact with facility's sensors using control devices. Such a natural response system to any changes in indicators at the enterprise notifies staff about accidents and problems, analyzes the efficiency of equipment use, assesses the state of the environment and the volume of waste emissions. Integration into the production of intelligent and IoT systems will increase the efficiency, accuracy and flexibility of process control at large enterprises, the quality of products and will allow to partially automate production processes, which will significantly reduce labor costs. Thus, IoT networks can be used by various farms [1], mines [2], factories [3], and environmental monitoring organizations to automate all processes of collecting, storing, and processing necessary information.

Most often, to implement such networks, a lot of used sensors are connected via wireless sensor network. These networks are based on IoT devices, which represent sensors connected to microprocessors with wireless access and a limited battery. Because of this, inefficient power consumption is the main problem of these networks, which reduces the network lifetime. To optimize power consumption, IoT specialists have created various algorithms that allow analyzing and managing sensors in the network, thereby increasing the lifetime of the wireless network. The results of their processing during simulations demonstrate their effectiveness with various network parameters.

1. Materials and methods

1. Objective

The primary purpose of this work is to find the optimal implementation of particle swarm optimization method and routing protocols and schemes for solving the problem of decreasing energy consumption in different topologies of wireless sensor networks by developing the package of software for designing network structures and process simulation of various optimization methods.

1. Tasks

The primary tasks are necessary to achieve the given purpose:

- Research various WSN routing schemes and protocols, existing simulation software, optimization methods, and energy consumption model.
- Selection of the most effective and development and testing of their implementations.
- Development of cross-platform application with graphical user interface for interacting with simulation parameters and network topologies.

- Testing of various kinds of network topologies and simulation parameters and compassion of results.

1. **Methods**

    1. Programming languages

Python version 3.8 programming language has been selected for software development. Python is an object-oriented programming language with dynamic typing and automatic memory control. It has certain advantages in comparison with other popular programming languages:

- Cross-platform.
- Many libraries and modules for mathematical operations and methods.
- Code reading and writing are significantly simplified.
- Many tools for graphical user interface designing.

Of course, this programming language has disadvantages. The main of them:

- Overall, the Python programming language is slower than other popular languages.
- Because of the flexibility of data types, the memory often exceeds this value in other languages.

Considering that in this work a cross-platform desktop application with a graphical user interface and many mathematical operations is being created, Python is a great choice to solve this problem. The limited speed of the program will be solved by the usage of the multiprocessing module. The memory overflow problems will be solved by the removal of the links on an unused object in the debug stage.

1. Libraries and frameworks

- Kneed is the implementation of the "knee locator" method in Python programming language. This library allows calculating the optimal number of clusters in a wireless sensor network based on the device's position in space.
- Matplotlib is the graphical library for displaying two-dimensional graphics. It displays the device state during the simulation process and draws the result charts.
- Numpy is the library used for performing mathematical operations on multidimensional arrays. Most used libraries, modules, and frameworks apply the data structures exactly from these libraries.

- Scikit-learn is one of the most popular libraries for machine learning and data science in Python. It contains implementations of many algorithms and methods including cluster-analysis algorithms, such as K-Means.

- Scikit-fuzzy is the specific toolkit for working with fuzzy logic based on the scientific calculation's library SciPy. This library contains the Fuzzy C-Means clustering method used in this paper.

- Pyswarms is the swarm intelligence processing library that has different implementations of the particle swarm optimization method. This document uses a binary and discrete implementation of this method which allows for optimizing a given fitness function by the binary array corresponding to the state of the device in a cluster or network.

- Tkinter is the classic python graphical interface toolkit. Significantly, this library is cross-platform, and each graphic element will correspond to the operating system on which the software is running. This library may include the Matplotlib widget as a separate interface element.

- Dijkstar is the library which represents the implementation of the Dijkstra algorithm for Python programming language. MTE (Minimal Transmission Energy) routing protocol uses this algorithm to find the shortest way between the node and the base station.

1. **Topologies and traffic routing methods**

Developed software supports classic and cluster traffic routing methods.

Classical methods of traffic routing:

- Direct communication. This routing scheme represents a binary tree whose units are the nodes of the wireless sensor network. Each of these nodes is connected to the base station directly (Fig. 1).

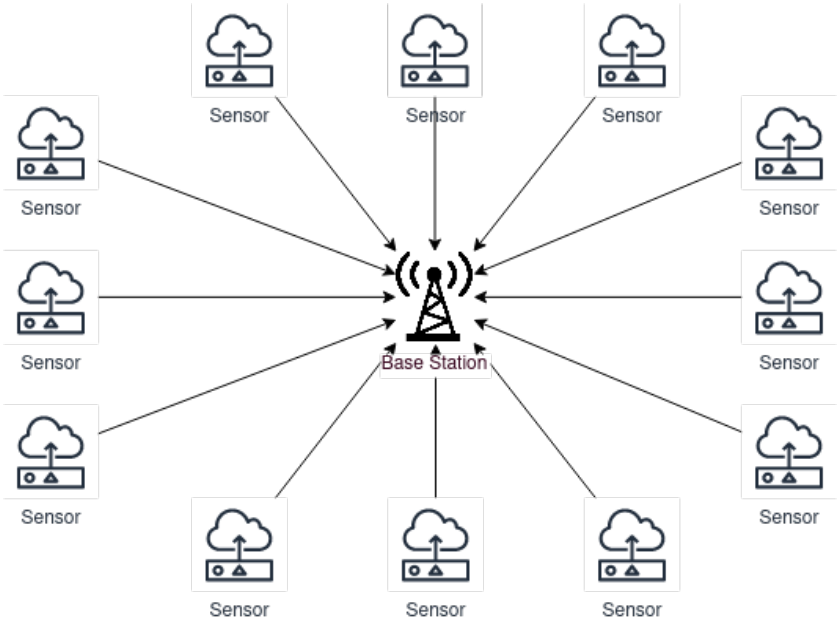

Figure 1. Direct communication

- MTE routing protocol uses the Dijkstra algorithm to find the shortest way between each node and base station through the nearest nodes [4]. To use this routing scheme it's necessary to transform a given network in graph representation (Fig. 2).

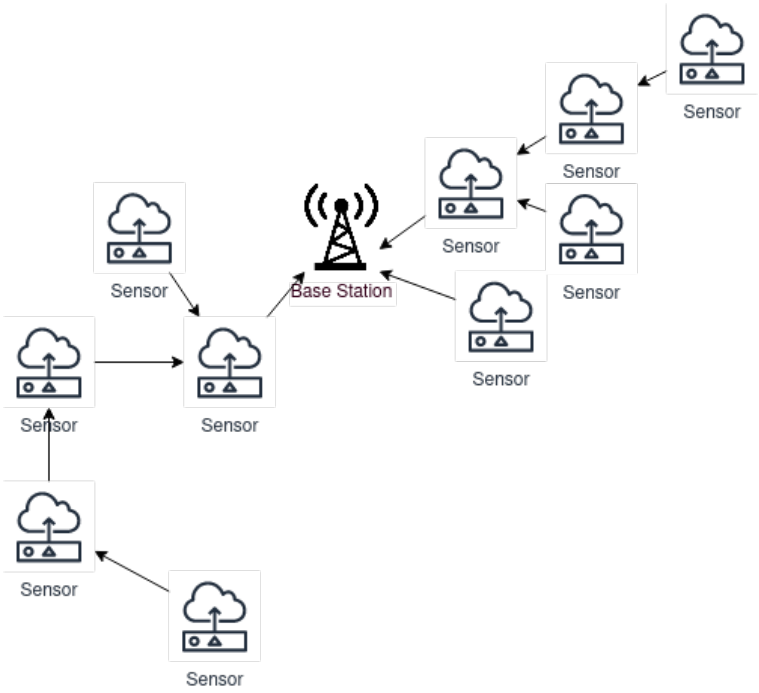

Figure 2. MTE.

Cluster methods of traffic routing:

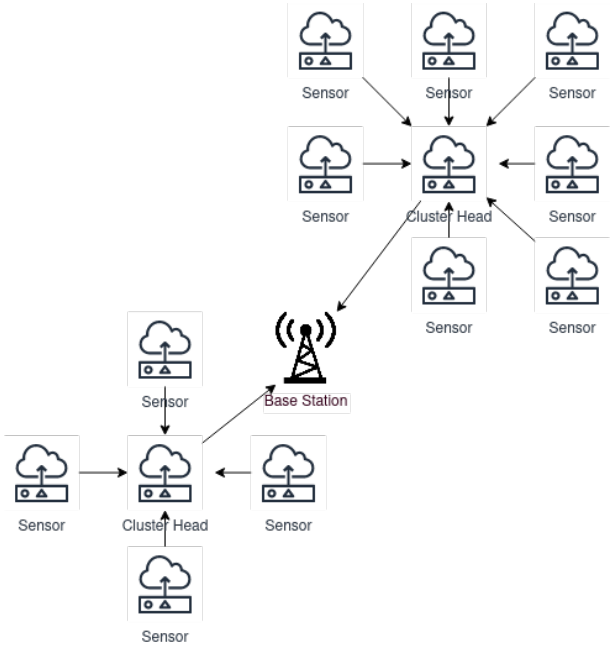

Figure 3. Cluster topology.

- LEACH (Low Energy Adaptive Clustering Hierarchy). This routing protocol chooses cluster heads based on a given probability which calculates for each node. All of the devices in the network send information to the nearest cluster head (Fig. 3). In this routing scheme the number of nodes in the cluster will be different in each iteration.

$$P_i = \ a * (\frac{E_i}{max(E)}) \ + \ \beta * (\frac{min(D_{\text{station}})}{D_{i,\ station}})$$

(1)
Where:
P_i is the probability of accordance given node to cluster head.
E_i is the amount of remaining energy on the given device.
$max(E)$ is the maximal remaining energy in the whole network.
$min(D_{\text{station}})$ is the minimal distance between the node and base station.
$D_{i,\ station}$ is the distance between the given node and the base station.

- Fuzzy C-Means cluster technique allows dividing a given set of IoT devices into a certain amount of fuzzy clusters. Fuzzy means that each node has the set of membership coefficients for each cluster in the network. Membership coefficients are calculated based on the distance to the centers of clusters. The cluster head's choice is based on the distance to the cluster center and

the remaining device energy. Network topology will be the same as the LEACH protocol (Fig. 3).

1. **Model of energy consumption**

Every IoT device in the network has three primary functions to ensure communication. All of these functions consume a certain amount of energy:

- Send byte-message - energy which is spent by the transmitter to send k bytes of information on distance D.

$$E = E_0 + E_1 * D^4{}_{\text{ij}} * k,\ D_{\text{ij}} > D_0$$

$E = E_0 + E_2 * D^2{}_{\text{ij}} * k,\ D_{\text{ij}} \leq D_0,$

(2)

Where:

E_0 is the energy which is consumed by the transmitter electronics.

E_1, E_2 is the energy which is consumed by the RF power amplifiers under different transmission conditions.

D_{ij} is the distance between i and j nodes.

D_0 is the coverage of the RF module.

- Receive byte-message - energy which is consumed by the receiver to receive k bytes of data.

$$E = E_r * k$$

(3)

Where E_r is the energy which is consumed by the receiver to receive one byte of data.

- Aggregate all of the received data - energy which is consumed by the device electronics to aggregate k bytes of received data to send it to the base station.

$$E = E_A * log(k)$$

(4)

Where E_A is the energy which spends to aggregate a minimal amount of bytes of data.

1. Sleep scheduling system and particle swarm optimization

Also, systems of node state managing using to optimize energy consumption and increase network lifetime, particularly the sleep scheduling systems. Different numerical optimization methods are used to calculate which node should go to a sleep state in each iteration. Swarm intelligence methods are used quite effective in this case [5-7]. This work presents the use of one of the most popular and effective techniques - PSO (particle swarm optimization).

Particle swarm optimization is the numerical optimization method based on the social behavior of the particle population. Each particle has a specific position and velocity in a given space. Particle movement is determined by the best positions of the whole population. The fitness function is used for calculating the best local and global positions for each node in the network.

It's necessary to compose the correct fitness function for effective optimization processing [8]. Such a function must set up the sleep state in each node in the sensor network. Each fitness function has several main parts, each of which considers one of the variables that affect energy consumption: remaining energy, distance to the base station or specific node, device load, interference and other variables multiplied by a definite coefficient that characterizes the weight of this variable. This paper presents fitness functions that take into account only energy and distance. In most cases, this is sufficient, but when using different types of touch devices, when placing sensor devices in a premise, you should think about considering additional variables. For different network topologies, functions are calculated differently:

- Direct Communication. It's clear that in this routing scheme, we should put into sleep mode more often the nodes with the least residual energy and located at the greatest distance from the base station.

$$f(m) = \alpha * (\frac{\sum_1^N E_i \text{ if } m_i = 0}{\sum_1^N E_i}) + \beta * (\frac{\sum_1^N D_{i,\ station} \text{ if } m_i = 0}{\sum_1^N D_{i,\ station}})$$

(5)
Where:
α, β – given constants.
m – input binary vector.
$D_{i,\ station}$ – the distance between the base station and node i.
E_i – node i remaining energy.
N – amount of the devices in the whole network

- MTE. This function is similar to Direct Communication, but in this topology, we need to put to sleep more often those nodes that are closer to the base station, since they are switching nodes.

$$f(m) = \alpha * (\frac{\sum_1^N E_i \text{ if } m_i = 0}{\sum_1^N E_i}) + \beta * (1 - \frac{\sum_1^N D_{i,\ station} \text{ if } m_i = 0}{\sum_1^N D_{i,\ station}})$$

(6)

Where:

α, β – given constants.

m – input binary vector.

$D_{i,\ station}$ – the distance between the base station and node i.

E_i – node i remaining energy.

N – amount of the devices in the whole network.

- LEACH, Fuzzy C-Means. In the cluster topology, the part with the variable distance also changes. In this case, we need to switch the nodes in sleep mode that are the farthest from the cluster's centroid since they are the farthest to send packets to the head node.

$$f(m) = \alpha * (\frac{\sum_1^N E_i \text{ if } m_i = 0}{\sum_1^N E_i}) + \beta * (1 - \frac{\sum_1^N D_{i,\ centroid} \text{ if } m_i = 0}{\sum_1^N D_{i,\ centroid}}))$$

(7)

α, β – given constants.

m – input binary vector.

$D_{i,\ centroid}$ – the distance between the cluster center and node i.

E_i – node i remaining energy.

N – amount of the devices in the whole network.

The constants α, β are determined experimentally, in this work they are taken as 0.7, 0.3

1. **Graphical User Interface**

The graphical user interface consists of several main parts (Fig. 4-6):

- The main window of the program represents combining the side menu and the visualization module with several interaction graphical elements.
- The side menu is the graphical module used for saving, loading, creating wireless sensor networks and setting up the simulation parameters.
- The network and charts visualization module represents a graphical interface element inherited from the Matplotlib library model for displaying it in the Tkinter widget.

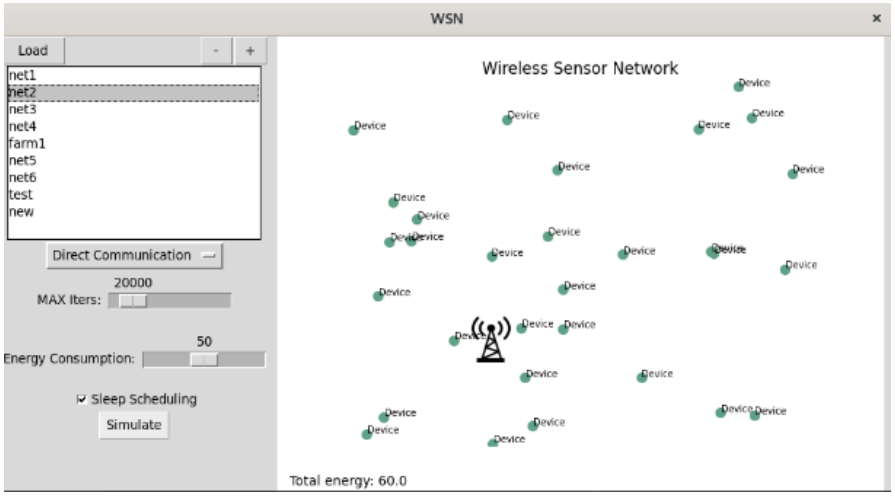

Figure 4. Main window – network choice.

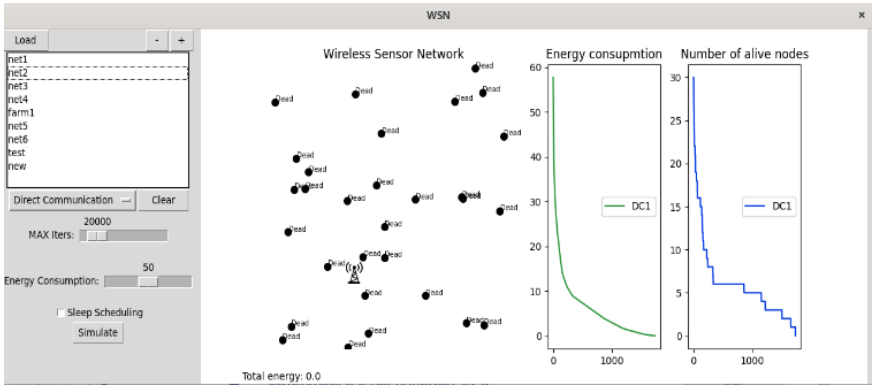

Figure 5. Results display.

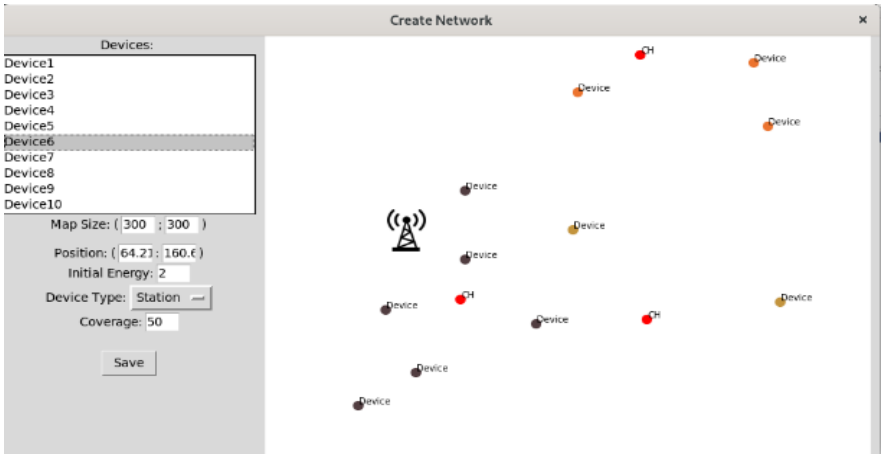

Figure 6. Map creation module.

1. **Testing**

The developed software has tested on two specific topologies of the wireless sensor networks for an independent comparison of optimization methods. The first topology has generated randomly, while the second represents the WSN topology for usage in the smart farm case.

- Randomly generated topology. This network was generated randomly and represents the set of the forty devices located within a space of 150x150 meters. The base station is located in the center of the map (Fig. 7).

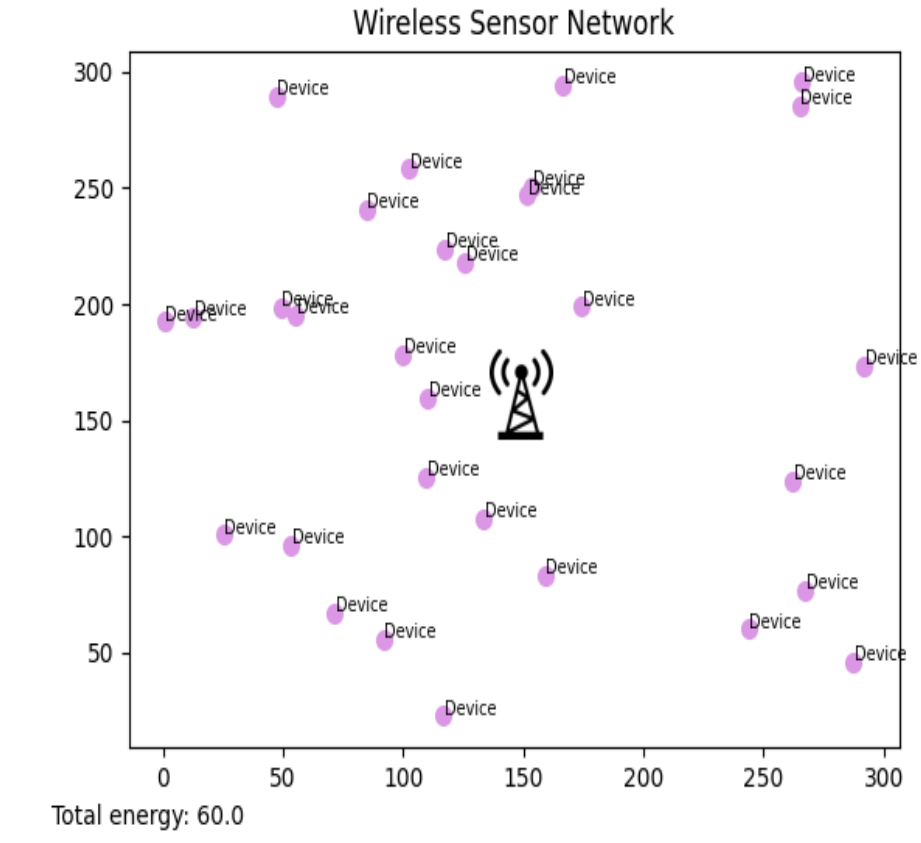

Figure 7. Randomly generated network topology.

- Smart farm. One of the possible applications of the wireless sensor networks is the monitoring of the soil state, temperature, and air humidity [1] in the greenhouse in agricultural industries (Fig. 8).

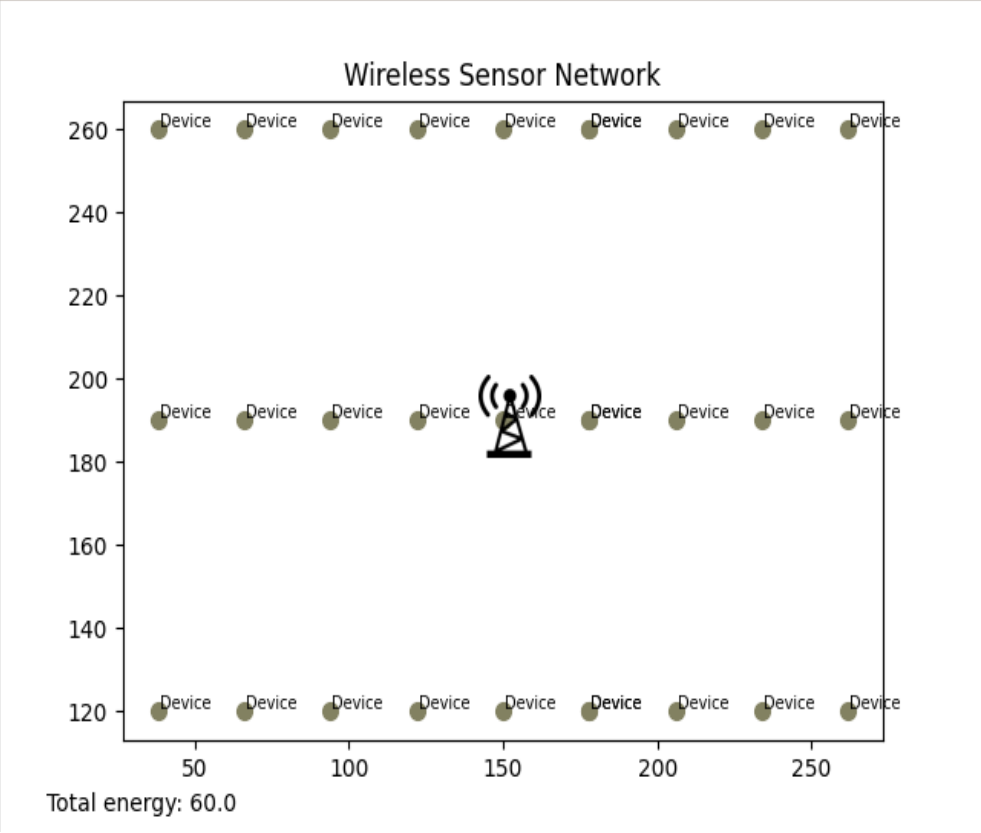

Figure 8. Smart farm network topology.

1. **Results**

After testing all of the routing methods for two given topologies with sleep scheduling systems and without and evaluating the results of processing presented simulations on charts, we can conclude about the effectiveness of usage particle swarm optimization methods in the case of nodes management systems in each routing method. It's clear that effectiveness of application of the sleep scheduling system in routing methods based on cluster topology is less than 5%. Nevertheless, resulting charts are showing that decrease of the energy and amount of remaining active nodes is significantly slower using the sleep scheduling system, compared to cases in which this system is not using. This indicates that this system is also quite effective in LEACH, Fuzzy C-Means routing methods. The results of processing LEACH simulation for both network topologies are presented in Figures 9-10. The first line on each chart is associated with simulation without the PSO sleep scheduling system, then the second with simulation with the system.

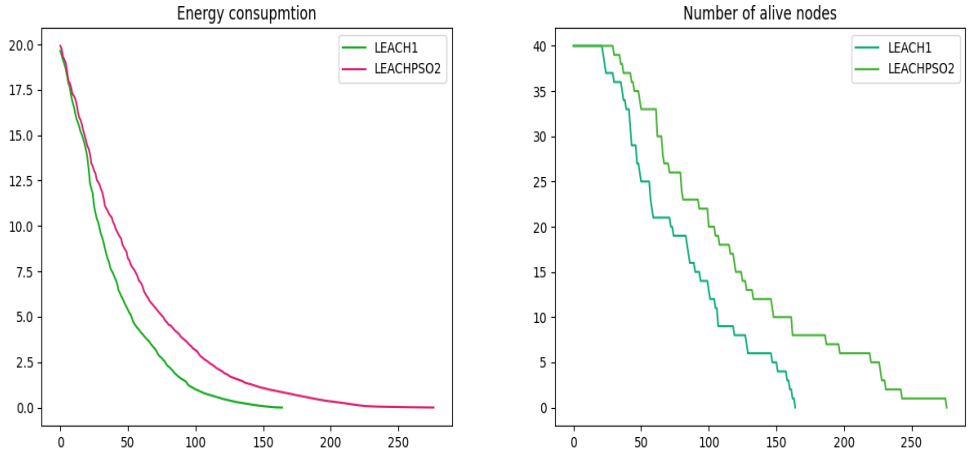

Figure 9. Randomly generated network LEACH simulation results.

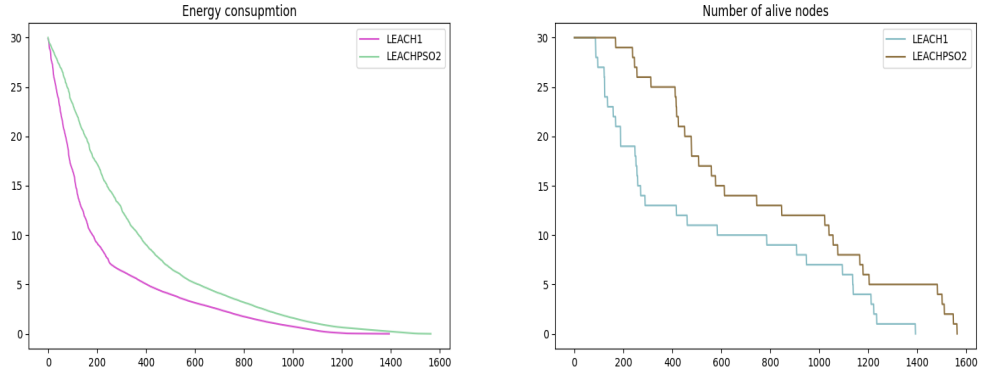

Figure 10. Smart farm network LEACH simulation results.

1. **Discussion**

For the design of wireless sensor networks, various ready-made solutions allow you to simulate different topologies of wireless networks.

- CupCarbon [9] is a wireless sensor network simulation software most used for smart-city network design. This program has a graphical user interface (Fig. 11) that allows you to set up the whole network and nodes separately. This software does not provide routing protocols and node state management systems, but it's can be implemented by programming the behavior of the nodes in an embedded script language. The simulator is written in Java programming language and is cross-platform software.

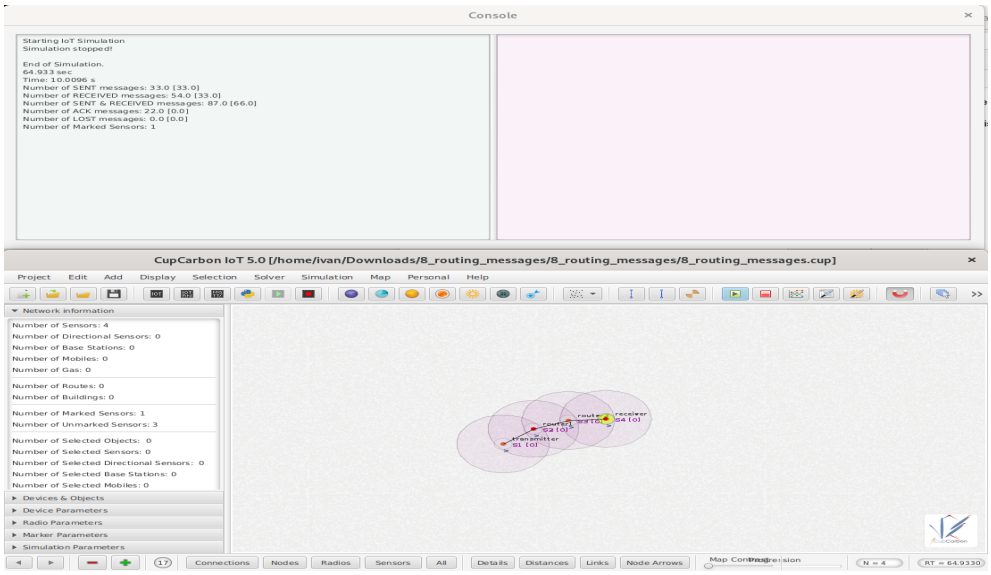

Figure 11. CupCarbon GUI.

- darolt/wsn [10] - a program for simulating wireless sensor networks without a graphical interface, which lets you set simulation, optimization parameters, and network topology through a specific configuration file. However, the program does have not the ability to fine-tune each node separately. Results of the simulation will be displayed after processing as Matplotlib charts (Fig. 12). This software provides different routing protocols and node state management systems which significantly reduce power consumption in the network. Also, it's written in Python and C++ programming languages and in general, the software is cross-platform.

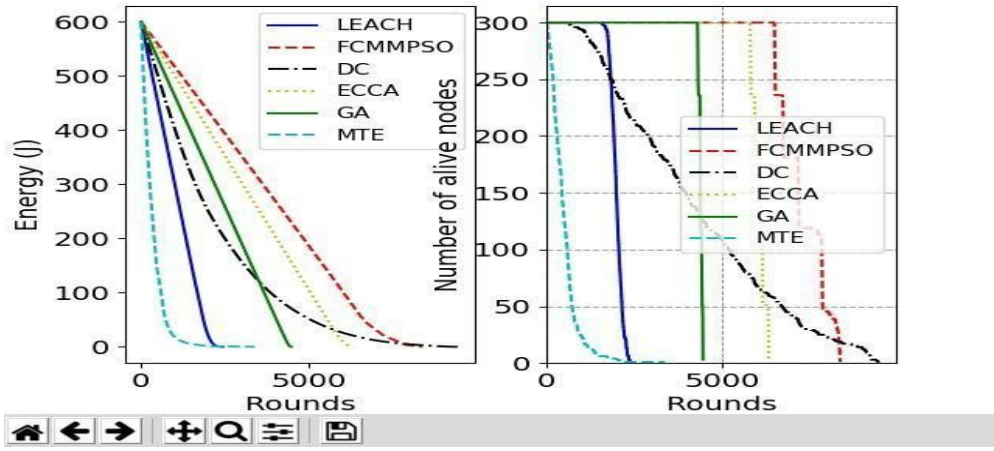

Figure 12. Results of darolt/wsn simulation.

- TRMsim [11] is a wireless sensor network simulator based on reputation and trust models. The software is distributed in the main for Windows operating systems but there is a small possibility of running on other operating systems. The program has a graphical user interface (Fig. 13) that displays the current state of the network at each time during the simulation. However, there is also no ability to set up each node separately.

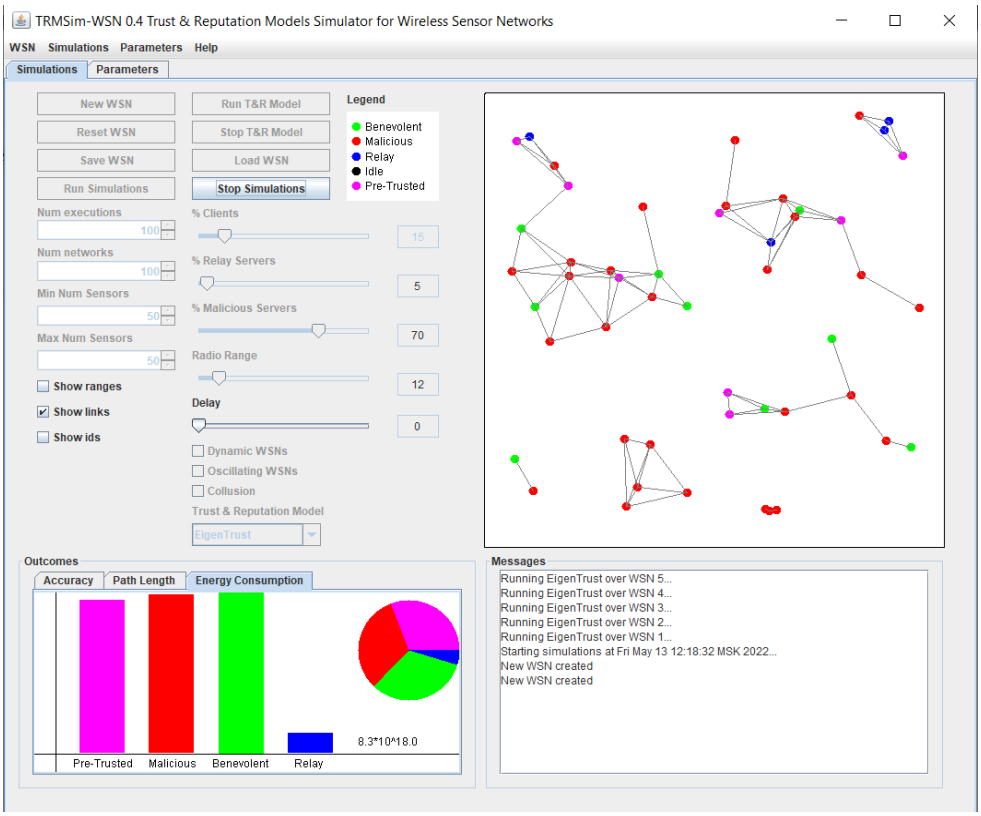

Figure 13. TRMSim GUI.

As a result of the research of existing software products, their limitations found and the main functional benefits of the wireless sensor network simulators:

- The presence of graphical user interfaces with the ability to create network topologies and display the network state during the simulation.
- The possibility to set up the separate nodes, whole network and simulation parameters.
- The availability of various routing methods and optimization techniques.
- The presence of node state management systems, such as sleep scheduling systems.

- Cross-platform.

Each of the presented benefits evaluations is based on the program results and the documentation provided with the software. The results had plotted in Table 1 with distribution by the presented criteria. From the results, it's clear that each of the presented above software has from 2 to 3 benefits while the developed software has all of them.

Table 1. Test results of external software products.

Simulator	Graphical User Interface	Ability to set up the network and nodes parameters	Routing protocols	Node state management systems	Cross-platform
CubCarbon	+	+	-	-	+
darolt/wsn	-	-	+	+	+
TRMsim	+	-	+	-	-
Developed software	+	+	+	+	+

1. Conclusion

Technologies in the field of the Internet of Things are swiftly developing every year, improving people's everyday lives and entire industries more and more. Developer specialists and researchers of IoT constantly develop and implement various routing methods, algorithms, schemes, and optimization methods which allow increasing efficiency of energy consumption and network control.

In this paper, the most effective and popular routing methods and topologies were researched and tested, particularly, classical and cluster routing techniques. The sleep scheduling system based on swarm intelligence was researched, implemented and tested for several network topologies. The cross-platform software package which represents the wireless sensor network simulator was developed. The program can create and fine-tune the network and each node separately. Also, it allows to set up one of the implemented routing and optimization methods. The program designed a user-friendly and intuitive graphical user interface thanks to which this software package can use not only by specialists in the wireless sensor networks field but also the people who have just started to study possible architecture solutions for the implementation of their own.

Improvement of the energy consumption model (accounting for interference and obstacles), appending of the various new routing algorithms, optimization techniques, and improvements to existing methods are possible in the future development of this software.

Acknowledgments.
Special thanks to my scientific director S.I. Yakimenko for help in writing this research work and for support in the development of the wireless sensor network simulator software at every stage of this work.

References

[1] Ardiansyah, D., Huda, A. S. M., Pratama, R. G. and Putra, A. P., *Wireless sensor network server for smart agriculture optimatization.* In IOP Conference Series: Materials Science and Engineering, 621(1) (2019, October) p. 012001.

[2] Kubrin, S.S., *Wireless sensor networks in coal mines*, Mining information and Analytical Bulletin (Scientific and Technical Journal), (S6), (2011), 159-165.

[3] Paavola, M., Leiviska, K., *Wireless sensor networks in industrial automation.* In Factory Automation. IntechOpen, 2010.

[4] Thomas, S., Gayathri, I.K. and Raj, A., *Joint design of Dijkstra's shortest path routing and sleep-wake scheduling in wireless sensor networks.* In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (2017), 981-986. IEEE.

[5] Jia, J., Chen, J., Chang, G. and Tan, Z., *Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm.* Computers and Mathematics with Applications, **57** (2009), no. 11-12, 1756-1766.

[6] Yu, C., Guo, W., Chen, G., *Energy-balanced sleep scheduling based on particle swarm optimization in wireless sensor network.* In 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, (2012), 1249-1255. IEEE.

[7] da Silva Fré, G. L., de Carvalho Silva, J., Reis, F. A., Mendes, L. D. P., *Particle swarm optimization implementation for minimal transmission power providing a fully-connected cluster for the internet of things.* In 2015 International Workshop on Telecommunications (IWT) (2015), 1-7. IEEE.

[8] Wang, L., Wang, X., Fu, J., Zhen, L. *A novel probability binary particle swarm optimization algorithm and its application.* J. Softw., **3** (2008), no. 3, 28-35.

[9] Software CupCarbon (2015). Retrieved from http://cupcarbon.com/

[10] Software darolt/wsn (2017). Retrieved from https://github.com/darolt/wsn

[11] Software TRMSim-WSN (2008). Retrieved from https://sourceforge.net/projects/trmsim-wsn/