Indian Journal of Science and Technology, *Vol 11(13), DOI: 10.17485/ijst/2018/v11i13/115861,* April 2018
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Event-Driven Simulation for WSN Clustering Algorithm

Solmaz Salehian[1*] and Rozita Salehian[2]

[1]Department of Computer Science and IT, Oakland University, Rochester, MI, 48305, USA; ssalehian@oakland.edu
[2]School of Engineering, Islamic Azad University of Malayer, Malayer, Hamedan, Iran; salehianrozita59@gmail.com

Abstract

Objectives: The reduction of energy dissipation and the avoidance of unbalanced energy consumption have been always considered as crucial issues in WSN. However, evaluating the solutions for these issues requires the agility of performance analysis tool. **Methods:** This research presents an effort that encompasses the development of a general-purpose programming language based discrete-event simulation to analyze specifically energy consumption in WSN. The Energy-aware Unequal Clustering Algorithm is used as a benchmark representative for the spectrum of WSN energy algorithms. The same domain with the aim of ensuring fairness is used in this study. The derivation of events, the event schedules and the time advancing mechanism are explicitly deliberated to enable the reproduction of the tools for extended cases. High emphasize is placed on the validation of the developed tool. The number of alive nodes and the distribution of dead nodes over the network are used as performance metrics to validate the developed tool. **Findings:** Analysis of the results has substantiated the proposed simulator as a member of the spectrum of analysis tools. **Applications:** As choosing simulation in the dynamic environment is challenging, thus, generic DES simulators is a good option, which can be applied in different domains and algorithms.

Keywords: Clustering, Discrete Event Simulation, Network Lifetime, Performance Analysis, Wireless Sensor Network

1. Introduction

There has been a surge in the impact of Wireless Sensor Networks (WSNs) especially with the momentum gain of the Internet of Things (IOT)[1,3]. However, there exist the challenges of striking a balance between the WSN node portability and the reliability of the source of energy to enable mobility. Issues such as the reduction of energy dissipation and the avoidance of unbalanced energy consumption are crucial issues in WSN. Finding the solutions for these issues requires the agility of performance analysis tool because real deployment is almost infeasible or impractical[4-9]. Performance analysis tools for WSN are provided via simulators, emulators and testbeds. Generally, two or more techniques are needed for a particular study to authenticate the acquired results. Simulation is essential for studying WSN related algorithm as deployment on real testbeds and experiments can incur high cost. There are several simulation tools for WSN with different layering, components and protocols. Furthermore, these presented tools vary in their attributes and properties subject to being either emulation or a hybrid in nature.

Emulators are different from simulators in a way that simulators model real environment and are software-based, while emulators run the same code on real platform. Thus, emulation uses the combination of both software and hardware to analyze performance. It is obvious that implementation of protocols in different platforms differ. Thus, even if two simulators are developed for the same protocol, there is a high possibility that the functionality is implemented differently from each other. The application domain creates a comparative platform which needs mapping and verification to enable definite determination of its reflection. The usage of the emula-

---

**Author for correspondence*

tion approaches, makes it possible to use the same source code for the simulator as well as the hardware platform. These distinct characteristics of various simulation environments create challenges for a clear recommendation to be given on the technique that is superior against others. Among some commercial and open source simulation tools for WSN, some more popular solutions among researchers are reviewed in following :

Network Simulator[11] (NS-2) is an open source discrete event simulator, supported by Defense Advanced Research Projects Agency and National Science Foundation. Open source simulators not only save the cost of simulation but also help users to modify the source code. However, NS-2 is not specifically designed for WSNs and special characteristic of WSN such as power consumption and bandwidth problem are not provided. Additionally, NS2 suffers from scalability problems as there is a limitation for the number of node.

OMNeT++[14] is a discrete event network simulator which is provided in a noncommercial and a commercial license. Powerful Graphical User Interface (GUI) in this tool makes trace and debug processes easier. However, it is not specifically used for WSNs and the number of available protocols is not broad.

J-Sim[2] is a discrete event network simulator which supports data diffusion, routing, localization, radio channel and power consumption in WSNs. A strong GUI in this tool provides users with the easier way of tracing and debugging. In comparison with NS-2, J-Sim is more scalable in terms of number of nodes.

QualNet[13] which is developed for wireless scenario can also simulate wired networks. It includes GUI, but it is not used widely because it is proprietary.

OPNET[10] is a commercial tool which has a large library of LAN network protocols, but it does not provide special characteristic for WSNs such as energy model and wireless mobility protocols. Consequently, the following main reasons contribute to the necessity of new simulator tools. Firstly, open source product such as NS-2, is a part of research work and has still undergone changes. This has a significant impact on program stability. On the other hand, commercial software such as QualNet and OPNET, demand purchasing a license and it is costly. Also, many of current popular simulators are general-purpose tools and are not designed specifically for WSN, so they have some constrains. In addition, due to the scale of the software and correlations of its components, full-fledge software is complex. These constraints or challenges can be overcome by developing a General-purpose Programming Language (GPL). The core element of this GPL is Discrete Event Simulation (DES). The details of this GPL based DES for analyzing the network lifetime and distribution of the dead node for WSN is presented in this study. DES involves the modeling of system and capturing events which occur through time. The time (i.e. clock representation) is advanced based on the chronological activation of events. The chronological selection of the events has significant influence on changing the proposed state in the system. The performance of DES is attributed towards the WSN operations that are indeed distinct from one process to another. Developing a DES involves a modeling that constructs a conceptual framework to describe the system process. In this research, the DES has formed the underlying platform of the proposed and developed analysis tool. The analysis of energy algorithm via the number of the alive sensor nodes and the distribution of dead nodes over the network as the primary performance metrics are enabled by this DES.

This paper is organized as following. In the subsequent section, the process of developing the proposed WSN routing simulator is presented. In Section 3, the assumptions used in the research are distributed. A detail discussion on the proposed and developed simulator and the respective performance metrics are extensively discussed in section 4. The validation of the simulation is presented in section 5. Section 6 concluded the paper.

2. EUCA-PSO

This section briefly presents the mechanism of the underlying algorithm that has been used. Energy-aware Unequal Clustering Algorithm (EUCA) serves as the foundation of the developed performance analysis tool in this work. The EUCA operates by utilizing the concept of rounds. Each round consists of two phases that is initialed by the set-up phase and is followed by the steady-state phase. At the start of the set-up phase, sensor nodes send their location information and respective energy levels to Base Station (BS). The BS finds the best Cluster Heads (CHs) between the CH candidates by using standard Particle Swarm Optimization (PSO) algorithm. The standard PSO is an optimization technique that has been inspired by the social behavior of bird flocking. The PSO optimizes a solution for a problem by trying a population, based on the movement within the search space and finds the best solution. This population is called swarm, and the candidate solution, called particle which has been chosen among the swarm. The PSO algorithm uses a cost function to navigate searching in the problem space. EBUC algorithm uses PSO algorithm to select an optimal group of sensors as CHs, and for cost function uses optimized parameters to form clusters in which unbalanced energy consumption is avoided.

For selecting the K best CHs among all possible candidates and partitioning network into the unequal clusters, the BS runs the PSO algorithm by applying the cost function in Equation (1). The K best CHs are chosen based on the minimum cost function with the aim of minimizing the intra-cluster distance between CHs and their nodes (Equation 2). Equation (3) is applied to quantify the optimizing of energy efficiency of the network. Clusters with unequal size which are smaller near to the BS are created by using Equation (4).

$$\text{Cost}\ (p_j) = \alpha_1\ f_1(P_j) + \alpha_2\ f_2(P_j) + \alpha_3\ f_3(P_j) \tag{1}$$

$$f_1(p_j) = \max k = 1,2,\ldots,k \sum \frac{d(n_i, CHp_{j,k})}{|Cp_{j,k}|} \tag{2}$$

$$f_2(p_j) = \frac{\Sigma_{i=1}^{N} E(n_i)}{\Sigma_{i=1}^{K} (CHp_{i,k})} \tag{3}$$

$$f_3(p_j) = \frac{\Sigma_{i=1}^{N} d(BS, CHp_{i,k})}{k * d(BS, NC)} \tag{4}$$

Where N is the number of nodes and NC is Network Center and CH $p_{j,k}$ is the candidate CH in the Particle j (j = 1,2,. . . ,Q). Having smaller clusters near to the BS saves more energy from intra-cluster communication for inter-cluster communication, so unbalanced energy consumption will be avoided and hot-spot problem will be addressed in this approach. At the end of the setup phase the clusters are and CHs are selected. Time Division Multiple Access (TDMA) for intra-cluster communication in order to avoid collision; and for inter-cluster communication uses energy-aware multi-hop solution is used to deliver data to the BS. At the steady state phase each member of clusters sends data to its own CH from one-hop routing based on the schedule that had received.

In this approach, each CH chooses a Relay Node (RN) which is responsible to forward collected data by the CH to the BS. This RN is chosen based on the energy level of nodes and their distance to the BS. Figure 1 shows the overview of this algorithm that clusters near to the BS are smaller and the multi-hop forwarding sends traffic from the CHs to the BS. Figure 2 shows a clustered network using the proposed simulation tool in this research. The same cost function and assumptions as benchmark are used to validate the acquired results from the developed simulation tools. It is obvious that the clusters closer to the BS, which is located outside of the network at (250,750), are smaller than the farther clusters with respect to the cost function used in the PSO. In this example, clusters are differentiated by different colors and red color represents the CH and each cluster has one CH.

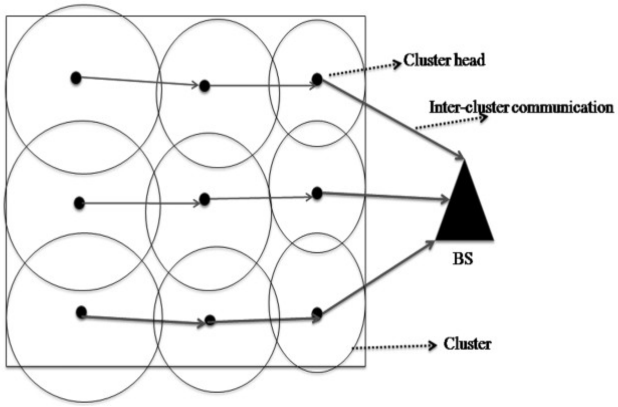

Figure 1. An overview of the unequal clustering protocol[5].

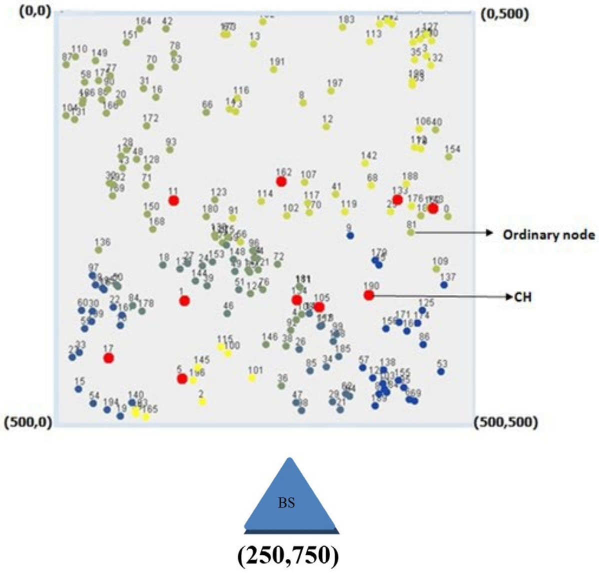

Figure 2. An example of clustered network by proposed simulation.

3. Assumption

3.1 Network Model

The assumptions are used to reflect a standardized model. In this work, the assumptions are based on the platform presented by[5] with the aim of ensuring fairness in our comparative analysis.

The WSN model has following properties :

- All sensor nodes have been uniformly deployed over the sensing field.
- All nodes are stationary and energy constrained.
- The nodes have power control capabilities to vary their transmitted power.
- The nodes can be either cluster heads or ordinary nodes.
- Data fusion is used to reduce the total data message sent.
- The BS is located outside of the network in order to show the effect of the protocol better.

3.2 Radio Energy Model

The nodes have power control capabilities to vary their transmitted power. Power control capability issued in order to minimize energy consumption in sending. The transmitter energy consumption is given by:

$$E_{Tx}(l,d) = E_{elec}l + E_{fs}ld^2; \quad d < d_0 \tag{5}$$

$$E_{Tx}(l,d) = E_{elec}l + E_{mp}ld^4; \quad d \geq d_0 \tag{6}$$

where,

l : refers to number bit of a message.

d : is related to distance between transmitter and receiver nodes.

d_0: is the threshold for distance. Regarding this threshold, either free space (fs) or the multi path (mp) model is used for transmitting (Equation 5 and Equation 6).

E_{elec}(nj/bit): is the energy used per bit to run the transmitter or the receiver circuit.

E_{fs} (pj/(bit.m^{-2})): is the energy used per bit to run amplifier when distance between the transmitter and receiver is less than the threshold.

E_{mp} (pj/(bit.m^{-4})): is the energy used per bit to run amplifier when distance between the transmitter and receiver is more than the threshold.

The receiver energy consumption is given by the following equation :

$$E_{RX}(l) = Eelecl \tag{7}$$

In the simulation we assume that information is sent from nodes to their own CHs. Then each CH compressed the received data to k bits regardless of the number of nodes in each cluster. The energy cost for data fusion in our simulation sets to 5nj/bit (ED=5 nj/bit).

4. Proposed Performance Analysis Tool

DES first constructs a conceptual framework to describe a system. The core of this system in our implementation is formed based on the algorithm of[5], which used Matlab for analyzing the performance.

The validation of the proposed simulator is highly controversial. Thus, the relative performance metrics are applied to compute the deviation error for this tool. The general programming language, JAVA is used as the underlying language for the development of DES.

Table 1 provides the list of parameters used in the developed simulation. The same values and domain as[5] are deployed in order to ensure the validation of the proposed simulator. 200 nodes are randomly distributed over a 500 m×500 m network area. The data message size was fixed at 2000 bytes; with 50 bytes representing the length of the packetheader. The number of clusters is set to 5% of the total nodes, so K=10. The BS located outside of the network at (250,750).

Table 1. Simulation parameters

Parameter	value
Base Station located at	250, 750
Network area	500 m×500 m
Number of node	200
The data message size	2000 bytes+50 byte
The number of clusters	5% of the total nodes, k=10
The initial energy of nodes	0.5j
Data aggregation energy (ED)	5 nj/bit
E_{elec}	50 nj/bit
E_{fs}	10 pj/(bit.m-2)
E_{mp}	0.0013 pj/(bit.m-4)

The PSO algorithm parameters are considered as following: Q = 30 particles, $c_1 = c_2 = 2$ and inertia weight from w = 0.9 to w = 0.4. We set $\alpha_1 = 0.3$, $\alpha_2 = 0.3$ and $\alpha_3 = 0.4$.

The initial energy of the nodes is set to 0.5J. The efficiency of the proposed algorithm is examined by evaluating the network lifetime and solving hotspot problem which is achieved through unequal clustering mechanism and the inter-cluster multi-hop routing.

4.1 Components and Performance Metric

The components in this work consist of sensors which can be either a CH or an ordinary node. Sensors, which are assumed to be stationary and have energy limitation, are distributed uniformly over the sensing field. Another component is the fixed BS. The developed simulator enables analysis of energy algorithms via the number of alive sensor nodes, mortality rate of the nodes and distribution of dead nodes over the network as the primary performance metrics.

4.2 Simulation Parameters

The energy cost for data fusion and the radio model of sensor nodes are considered control parameters. These parameters serve also in validation of the developed simulator.

4.3 Simulation Structure

The design of simulator constitutes of two parts. The first part is the logic of algorithm, and the second is the components and functionality of the tool. The crust of performance analysis is in bridging the two components for diverse domains.

The main task in developing an authentic performance tool is to derive the events. Events change the statistical composition in WSN (i.e. packets).

In simulating the EUCA, elements such as nodes, CH, RN and BS play pivotal roles as the crucial indicators of statistical compositions. In this research, packets are considered as the major statistical composition, due to the fact that they are the primitive observatory elements in the performance metrics computing. The implemented simulator derives three main events, which can occur for four entities. The events are send, receive and forward and entities refer to the arbitrary nodes, CH, RN, and the BS. Figure 3 illustrates the sequence of events in EUCA algorithm which are indicated by alphabetic labels. Table 2 lists all possible events respectively for the first round in EUCA algorithm. Figure 3 describes all steps of implementation for the proposed simulator for diverse phases including network setup, setup phase and steady state phase.

- In network setup that the size of the network and the BS location are defined. Then nodes are deployed over network area with respect to the mode of node distribution. The assumption on the energy level for each node is also initialized in the network setup.
- In the Setup phase, when the network has been formed, all nodes send their information to the BS which encompasses two events: sending nodes' information to the BS and receiving information by the BS. These events are indicated as label 'A' in Figure 3. Subsequently, based on the logic of the EUCA algorithm, the BS runs the PSO algorithm and finds the best candidates for CHs. The BS then broadcasts results to all nodes, which is indicated as label 'B'. The details of all events are presented in Table 3.
- Steady-state phase is started with creating TDMA schedule by the CHs. Each CH then sends the schedule to all member nodes in order to avoid

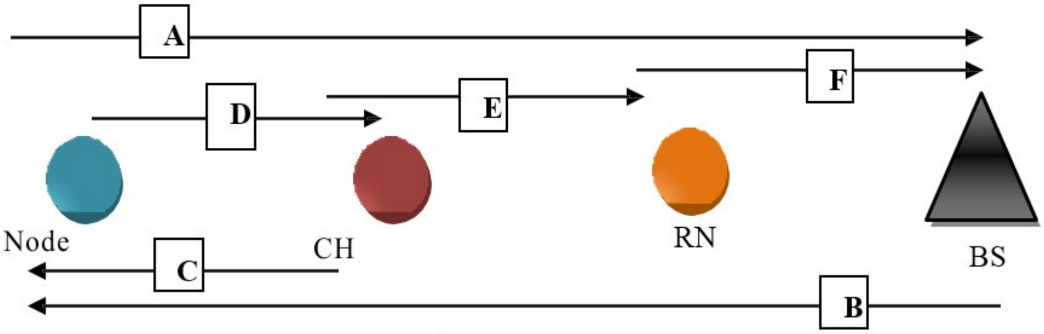

Figure 3. Events process.

Table 2. The sequence of events in the first round

Label of event	Event number	Description
A	Events 0	Nodes send info to the BS
	Events 5	The BS receives data from nodes and runs the PSO algorithm
B	Events 4	The BS sends info to the nodes
	Events 1	Nodes receive info from the BS
C	Events 8	CHs broadcast TDMA schedule to its own members
	Events 2	Nodes receive schedule from its own CH
D	Events 3	Each node sends its own data based on the received schedule time
	Events 9	The CHs collect data
E	Events 10	The CHs send data to its own RN
F	Events 7	The RNs forward data to the BS
	Events 6	The BS receives data from the RNs

Table 3. Details description of events

Event 0

```
For every node
{
Go to wake mode in order to send its information to the BS
Define the time that the BS receives information from nodes (based on their distance)
Decrease the transmit energy value for the node when it sends its information to the BS
Go to sleep mode
}
Call the next event
```

Event 5

```
{
PSO algorithm is run by the BS
Define the time that the BS broadcast information to nodes by considering PSO execution time
Call the next event
}
```

Event 4

```
For every node
{
Define the time that node receives information from the BS according to its distance
}
Call the next event
```

Event 1

```
For every node
{
Decrease the energy level for each node as it receives information from the BS
}
For every CH
{
Create TDMA schedule
Find the number of associate members to the cluster
```

```
    Consider max number of members as the required time for collecting data by each CH
    Devote equal time to members of each CH for sending their data
    Define the time that CHs create and send TDMA schedule to its members
}
Call the next event

Event 8

For every CH
{
    Set wake mode for all their members
    Decrease the energy level of each CH as it transmits TDMA schedule to its neighbors
}
Call the next event

Event 2

For every node
{
    Decrease the energy of the node as it receives the TDMA schedule
    Define the time that each node sends data to its own CH
}
Call the next event

Event 3

For every node
{
    Decrease the energy of the node for data transmission to its own CH
}
For every CH
{
    Define the time that CH receives and collects data
}
Call the next event

Event 9

For every CH
{
    Collect and fuse data
    Decrease the level of energy for the CH once it collects data from members
    Define specific node as the RN for each CH based on distance and the level of energy
    Define the time that CH sends data to RN by considering the required time for data fusion
}
Call the next event

Event 10

For every CH
{
    Decrease the level of energy for CH when it sends collected data to its own RN
}
For every RN
{
    Define the time that RN receives data from CHs
}
Call the next event
```

```
Event 7
For every RN
{
Decrease the level of energy for each RN once it receives data from CHs
Define the time that each RN needs in order to forward data to the BS by considering its distance to the BS
}
Call the next event
Event 6
For every RN
{
Decrease the level of energy for RN when it sends data to the BS
}
For every node
{
Define initial time for the next round (event type=0)
}
Call the next event
```

any collisions between nodes while collecting data. This process encompasses two events, first when CHs sending the TDMA schedule to the nodes and second when nodes receive its own schedule. These events are indicated by Label 'C' in Figure 2. According to the received schedule, each node starts sending data and CHs then fuse all the received data. these steps are shown by Label 'D' in the Figure 2. Label 'E' represents the event in which CHs send the fused data to its own RN. Forwarding data by the RN to the BS and receiving data by the BS are the last step of the first round. These processes are shown by label 'F'. After the first node nodes do not need to send their information about energy level and location to the BS again because the BS is calculated energy level of each node based on the first round. Therefore, the next round can be started by the PSO execution in the BS. Scheduler is responsible to select the next event which should be executed in the current round. The events are derived based on the process of the packets. The packet lifecycle through nodes, CHs and the BS define events. The scheduler selects events based on the assigned time associated to the events.

5. Experimental Results and Discussion

In this section, the simulator is evaluated and validated by two evaluation scenarios and subsequently their respective results are discussed. The objective of the evaluation is to verify the ability of the proposed simulator in creating unequal cluster by using the same PSO cost function[5] and distributing of the dead nodes over the network and network lifetime. The results show that our simulator is produced approximately same results as the benchmark and successfully is able to simulate the same behavior. Figure 4 shows the performance of the simulator in creating unequal clusters by using the same PSO cost function as EUCA algorithm. The objective of having smaller clusters near to the BS (located at (250,750)) is to address hotspot problem. Hotspot problem is created by the high number of dead nodes close to the BS. Referring to Figure 5 the simulation follows the same pattern as the benchmark in distributing the dead nodes over the network and avoid hotspot problem. This has been achieved because CHs of smaller clusters near to the BS have fewer members to communicate and consequently save more energy for communication with the BS.

Figure 6 illustrates that the developed simulator performs and produces approximately similar results as the benchmark. The developed simulator is successfully been able to simulate the behavior of the network lifetime. Two senarisos of the proposed tool in Figure 7 are presented in order to show the accuracy of the tool. Based on our assumption, the number of nodes is considered 200 and 10 nodes are chosen as the CHs. The acquired results show same trend as the benchmark with less than 10% margin. By considering the

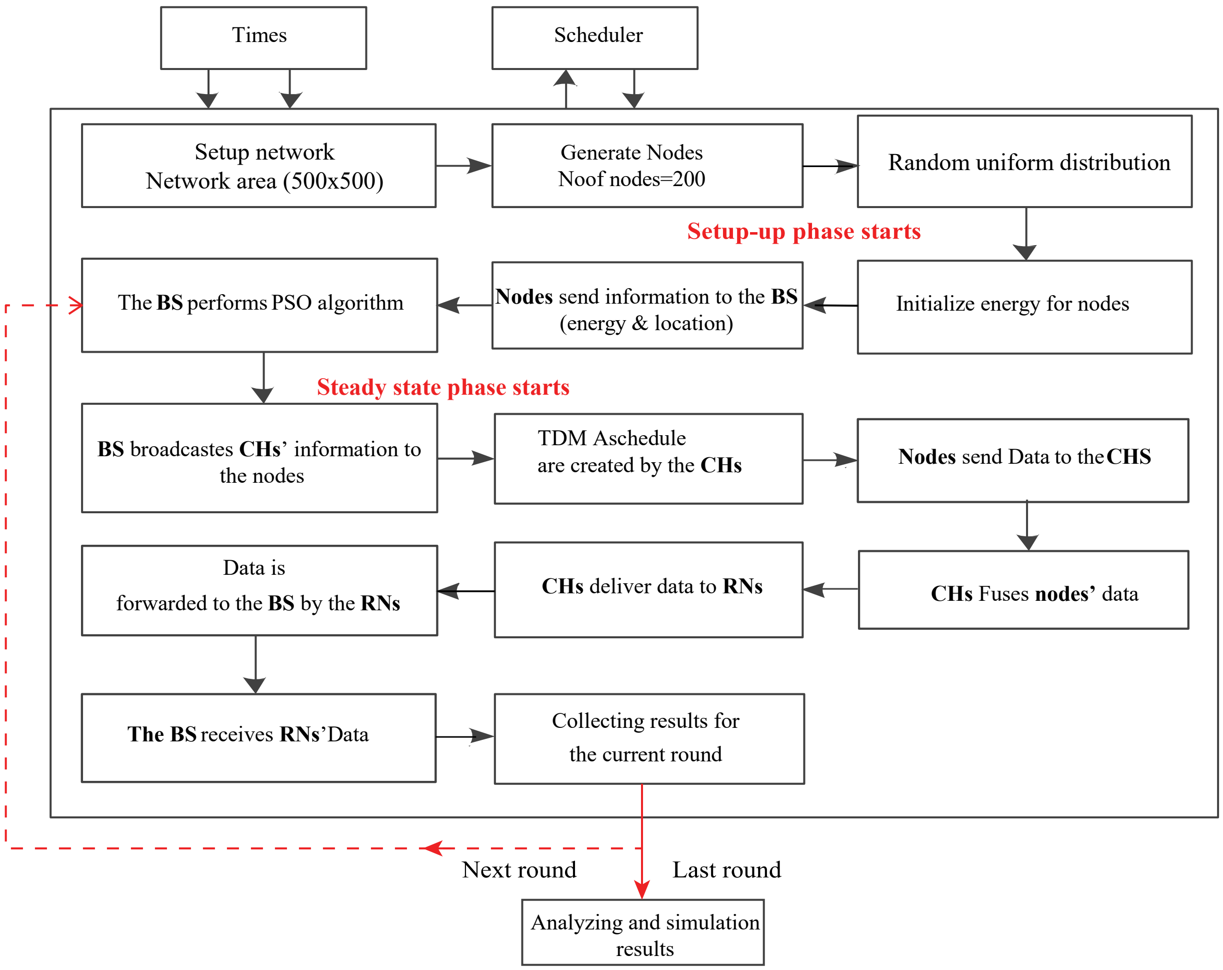

Figure 4. The implemented simulation structure.

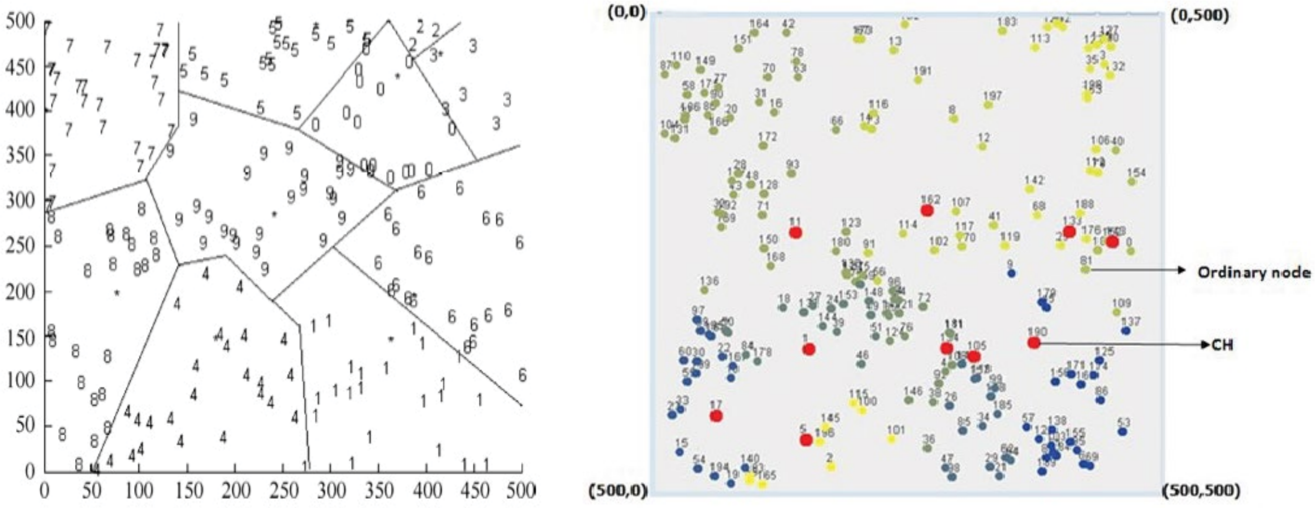

Figure 5. Comparison of clustering in Benchmark and the developed simulator.

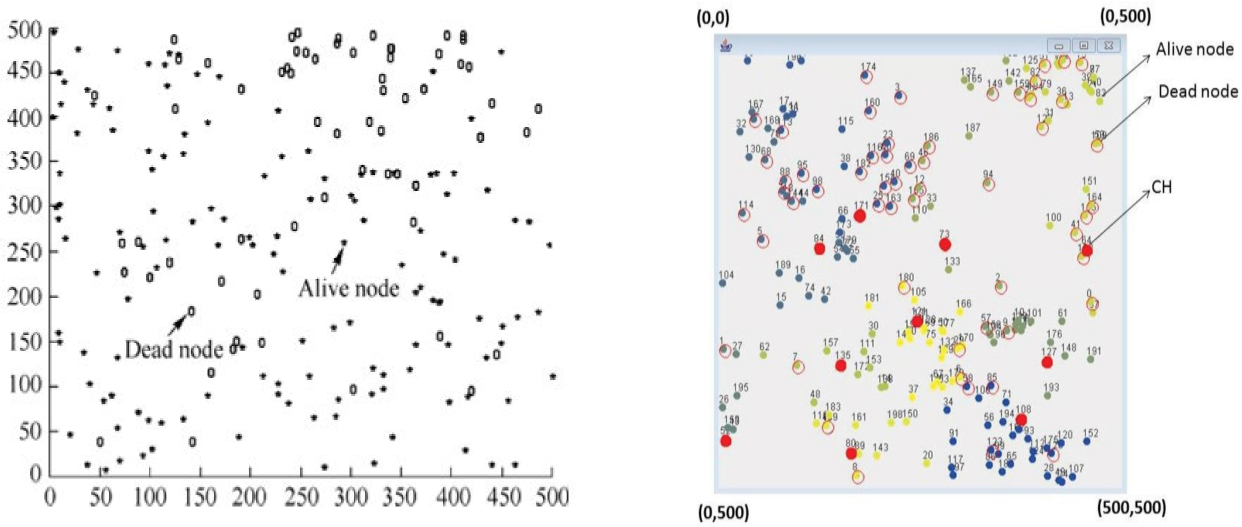

Figure 6. Comparison of dead nodes distribution in Benchmark and the developed simulator.

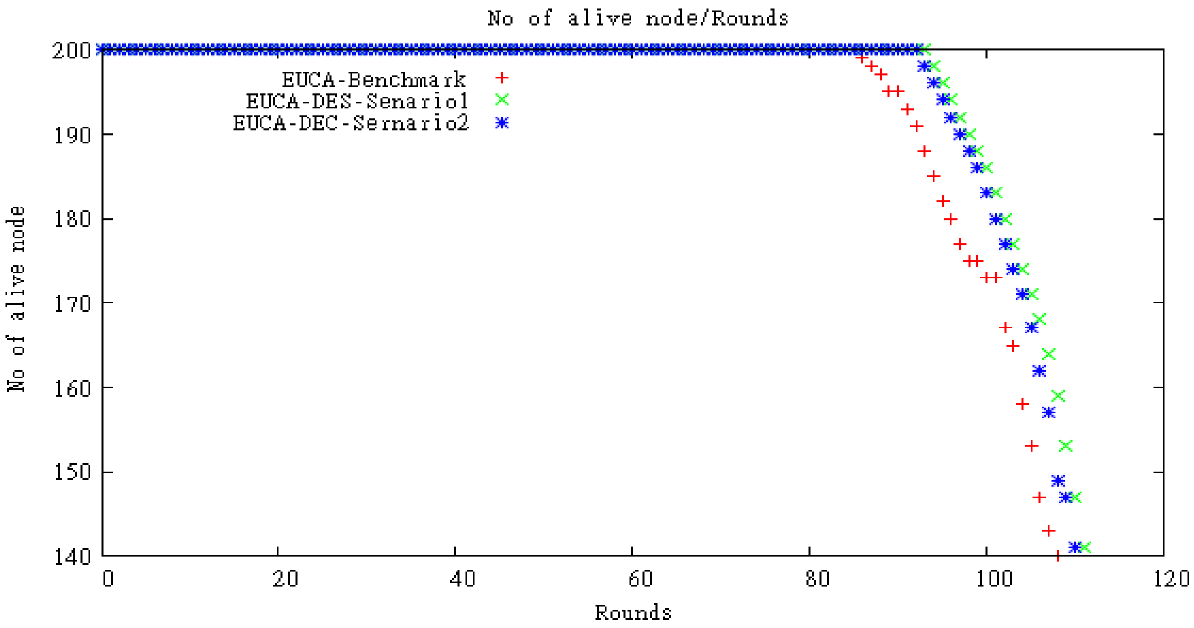

Figure 7. The comparison of number of alive node until 30% of nodes die for the benchmark and the development stimulator.

random properties in the simulations, this percentage has been accepted as being within an acceptable level of difference.

6. Conclusion

There is a wide-spectrum of simulators to cater the needed tools for performance analysis such as NS-2 and commercial tools such as OPNET and the QualNet Modeler. However, each of which has their own respective limitations. They have integrated components, complexity and licensing cost. These constraints or challenges can be overcome by developing a DES tool. In this paper, a GPL based DES has been developed to analyze both the network lifetime and distribution of the dead node for WSN. To validate and verify the proposed simulator, we consider two scenarios in order to show the accuracy of the tool and the result in both scenarios show the same behavior as the benchmark and successfully fulfill the requirements. The selection of a simulator is a controversial issue in WSN as it depends on different factors. There is a high possibility that the deployed environment to change rapidly. Thus, flexibility is considered as an important factor. There is no clear-cut decision about the simulation in this dynamic environment. Thus, generic DES simulators can be a good option and it can be applied in different domains and algorithms. In addition, it addresses some possible constraints in the current simulations such as the number of nodes.

7. References

1. Esteban Egea-Lopez, Javier Vales-Alonso, Alejandro S Martinez-Sala, Pablo Pavon-Marino, and Joang García-Haro. Simulation tools for wireless sensor networks. In Summer Simulation Multiconference, SPECTS, pages 2–9, 2005.
2. Jennifer Hou. Javasim, 1999. 3.
3. Salehian S, Subraminiam SK, Salehian R. Energy Aware Routing Strategies for an Evolving Wireless Sensor Network: A Survey. InHandbook of Research on Wireless Sensor Network Trends, Technologies, and Applications 2017 (pp. 196-236). IGI Global.
4. Muhammad Imran, Abas Md Said, and Halabi Hasbullah. A survey of simulators, emulators and testbeds for wireless sensor networks. In Information Technology (ITSim), 2010 International Symposium in, volume 2, pages 897–902. IEEE, 2010.
5. Chang-Jiang Jiang, Wei-Ren Shi, Xian-lun TANG, et al. Energy-balanced unequal clustering protocol for wireless sensor networks. The Journal of China Universities of Posts and Telecommunications, 17(4):94–99, 2010.
6. R Kennedy. J. and eberhart, particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks IV, pages, volume 1000, 1995.
7. Mahmoud Shuker Mahmoud and Auday AH Mohamad. A study of efficient power consumption wireless communication techniques/modules for internet of things (iot) applications. Advances in Internet of Things, 6(02):19, 2016.
8. Mehdi Mekni and Bernard Moulin. A survey on sensor webs simulation tools. In Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on, pages 574–579. IEEE, 2008.
9. Danial Naghshbandi, Shamala Subramaniam, and Saeed Rahimi Moghadam. An event-driven simulation for highest urgency first (huf): A latency and modulation aware bandwidth allocation algorithm for wimax base stations. Wireless personal communications, 69(1):39–52, 2013.
10. Riverbed. Opnet, 1986. crossref. Date accessed 09/20/2017.
11. The Network Simulator. Ns2, 1889. http://nsnam.sourceforge.net/wiki/index.php/Main_Page/.
12. TR Sreedevi and Mary Priya Sebastian. A classification of the debugging techniques of wireless sensor networks. In Advances in Computing and Communications (ICACC), 2012 International Conference on, pages 51–57. IEEE, 2012.
13. Scalable Network technologies. Qualnet, 2008. crossref. Date accessed 09/20/2017.
14. András Varga. Omnet++, 2001. crossref. Date accessed 09/20/2017.