

Mobi-Sim: An Emulation and Prototyping Platform for Protocols Validation of Mobile Wireless Sensors Networks

Omina Mezghani^{1,2}, Pr. MAHMOUD ABDELLAOUI^{2, *}

¹ ISI, National Engineering School of Sfax (ENIS), Tunisia

² WIMCS, National Engineering School of Electronics and Tele-Communication (ENETCOM), Tunisia

ARTICLE INFO

Article history:

Received: 10 December, 2016

Accepted: 14 January, 2017

Online: 28 January, 2017

Keywords :

Mobile WSNs

WSN simulator tools

Mobi-Sim

Clustering

Routing

ABSTRACT

The objective of this paper is to provide a new simulator framework for mobile WSN that emulate a sensor node at a laptop i.e. the laptop will model and replace a sensor node within a network. This platform can implement different WSN routing protocols to simulate and validate new developed protocols in terms of energy consumption, loss packets rate, delivery ratio, mobility support, connectivity and exchanged messages number in real time. To evaluate the performance of Mobi-Sim, we implement into it two popular protocols (LEACH-M and LEACH sink-mobile) and compare its results to TOSSIM. Then, we propose another routing protocol based on clustering that we compare it to LEACH-M.

1 Introduction

Wireless sensors networks represent a very interesting multidisciplinary field providing tremendous benefit for a number of domains in daily life. It is compounded by a set of sensor nodes characterized by a low-energy consumption in both hardware and software to maximize the network lifetime [1-2]. These characteristics make WSN suitable to be used in several applications like military, domotic, health, environment... In spite of its various usage in numerous applications, WSN confronts many issues like energy, self-organization, sensors localization, fault tolerance, security, mobility (since many applications require the presence of mobile entities in the network) [3]. So, WSNs field is under rigorous research to develop new algorithms, techniques and protocols to make more efficient and reliable the mobile WSN. In fact, nodes are often expected to operate for long periods of time. So, the design of protocols for mobile sensors network requires adequate methods for efficient data routing and minimal energy consumption. Then, researchers need to test and validate their proposed approaches before their implementation in a real system. So, simulation is the common way used by re-

searchers to test and validate their developed prototypes in various forms (energy, connectivity, QoS ...) before any implementation. There are many simulation tools, out of which they are dedicated towards traditional networks and just have been adapted to be used with WSN, other support only fixed nodes in the network and are unable to model mobile nodes behavior. Therefore, implementing new protocols under these simulators is often difficult and they cannot model all the required performance criteria (energy, loss packets rate, connectivity ...).

This paper presents a new simulator 'Mobi-Sim' that emulate a sensor node at a laptop in order to study the dynamic behavior of mobile nodes in real time without any scenario for mobile sensors movement programmed in advance. In fact, each laptop running the Mobi-Sim application can replace a node and behave as a sensor within a network. We can imagine that this ambition poses very difficult problems to solve at all levels in a surveillance system:

- The exchange of data in the network,
- The routing algorithms process,
- Battery level monitoring in real time,

*Corresponding Author: Pr. Mahmoud Abdellaoui, WIMCS-ENET'COM, (+216) 50 42 94 27 & mahmoudabdellaoui4@gmail.com

- The system reliability in terms of connectivity between moving nodes

To evaluate the performance of Mobi-Sim, we implement into it two popular routing protocols (LEACH-M and LEACH sink-mobile) to compare its result with those obtained by TOSSIM. In fact, in the previous work [4], we just implemented and tested the flooding protocol into our proposed simulator.

On the other hand, since many applications require the presence of mobile entities and nodes are often expected to operate for long periods, new challenges for WSN management are envisaged. So, the design of new routing protocols adequate for mobile sensor networks is very demanded. For this reason, we describe in this paper a new routing protocol based on clustering that supports mobility. We validate it by the developed Mobi-Sim simulator because the existing simulators do not model all the required performance criteria (energy + loss packets rate + connectivity + number of events).

2 Simulation Tools

In this section, we present an overview of the existing simulators for both real and non real time systems to find out their limits and their problems. These simulators are discussed according to different scales: network deployment & planning, data routing and performance metrics.

To overcome these simulator's limits, we propose a new simulator "Mobi-Sim" which answer at all expectations and all requirements of researchers and designers. Notice that protocol engineering requires an unavoidable phase which is the evaluation of the new developed protocols [5]. Several tools are offered for researchers and designers, each has its advantages, but also its disadvantages.

- **Ptolemy II** addresses the modeling, simulation and design of real time embedded systems. It offers different models of simulation paradigms such continuous time, data flow and discrete events. It is based on java packages. But, this simulator implies some limits in terms of scalability issues. In fact, simulation of several nodes remains a challenging problem [6].
- **IDEA** is a systemC simulator for WSN. It implements a power module for energy consumption modelisation. It can also model the delivery packet rate and transmission latency. But, it fails to simulate mobile sensor nodes in the network [7-8].
- **True Time toolbox** is a Matlab library dedicated to real-time system. It is a simulation tool for WSN. It proves its ability and flexibility to support different physical layers. It also provides a graphical environment that make easy the designing and the implementation of a variety of systems. But, in this simulation tool the step from simulation to production code is not large. Therefore, execution of experiments is time consuming [9-10].
- **MULE** is a hybrid simulator. It performs the execution of events in real time where physical motes are connected to PC. This simulator is dedicated for fixed WSN and cannot model mobile sensors behaviour. Therefore, this tool does not offer a graphical user interface [6].
- **COOJA** is a simulator mainly developed for wireless networks. It is useful when simulating sensors which are running the Contiki Operating System . COOJA is a cross-layer simulator. It operates at three abstraction levels. The networking level is the highest one, at which routing protocols can be implemented by users. The developed program code which is tested at the operating system level on COOJA could be run directly on Contiki OS without modifications. The third level is the machine code instruction. In this level, the emulation is done at a bit level [9]. Simulated sensor nodes can be from different type in terms of hardware and software aspects including sensors which don't use the Contiki operating system. But, it has some disadvantages. First, each node is simulated solely at one of the supported level. Second, cross-layer behavior exhibits increased granularity in simulation outcomes which is not available with TOSSIM [9-10]. Third, due to its extensibility, COOJA has relatively low efficiency. Therefore, it Supports a limited number of simultaneous node types, so it has to be restarted if this number exceeds the limit.
- **SENSE** is a discrete event simulator written in C++, written in CompC++ (an extension to C++). SENSE has modules for battery models, MAC Layer protocols, application, network and physical layer models. It also includes a set of SSR Protocols such as Self-Selective Routing (SSR), Self-Healing Routing (SHR) and Self-Selective Reliable Path (SRP). The iNSpect tool is used by SENSE to allow simulation visualization in order to analyze the simulation output and perform simulation animation showing message flows [11-12]. SENSE can be modified easily and is fast compared to NS-2. But, the need for direct integration into developed applications makes the use of NS-2 and SENSE unwanted. Therefore, SENSE don't support mobile nodes and cannot model mobility behavior of sensor nodes.
- **TOSSIM** is a TinyOS simulator. TinyOS is a sensor-based network operating system [13].It allows to simulate the sensor behavior (sending/receiving messages, information processing, ...) within a WSN. The TinyOS execution mode is directed by events and is adapted to discrete events simulator. However, TOSSIM did

not allow the execution of more than one application on the network and to estimate important behaviors including CPU and energy characteristics [14-15]. An extension of TOSSIM called **PowerTOSSIM** is added to model the energy consumption [16]. For the consumption values, the designers relied on the Mica2 type nodes. The designers relied on the different components consumption in a node according to their states. So, it is necessary to know the state of each component in a node during the simulation. Thanks to the simulation model based on TinyOS, the components state is known immediately since their changes correspond to events in TinyOS and thus in TOSSIM. However, several components in a sensor node are sometimes abstracted in TOSSIM by a single component [6]. Therefore, this tool does not offer an ergonomic user interface and does not support the mobility of nodes.

- **OMNet++** is a discrete simulator implemented in C ++. Getting started with this simulator is quite simple thanks to a clear design of the simulator. It also provides a powerful graphical user interface for debug animation and management. In addition, a framework for mobility has been added and it can be used to model WSNs. However, there is a serious lack of available protocols in the library compared to other simulators such as NS-2. On the other hand, there is a compatibility problem because researchers develop their problems separately which make its combination difficult and may lead a high number of bugs [14].
- **NS-2** is a discrete event simulator. NS-2 proposes four levels of abstraction which allow to adapt this simulator to different interests. Indeed, some will want low-level information, for example to study the effect of radio signals propagation in an environment while others study routing protocols and only need information at the network level. NS-2 was originally intended for wired networks, which certainly explains the simplicity of the radio propagation models in this simulator. Despite the different levels of abstraction, NS-2 is mainly used by researchers who are interested in routing protocols and/or medium access protocols. The open source NS-2 code allows each one to add his stone to the building. NS-2 research remains a very used simulator. However, the simulation is not real time; It is considered virtual. Therefore, it is used to simulate traditional networks and it is not targeted to WSN which make it behave differently than WSNs. In fact, packets format, MAC protocols, energy models and the sensing hardware are not the same with those found in most sensors. Another problem is the tracing system. In fact, it has a poor graphical support i.e. without GUI. So, users have to

manipulate text commands of the electronic devices [11]. The same for **NS-3**, it does not provide any graphical tool for analyzing or visualizing network simulation. Therefore, although the NS3 models are written in C++ like NS2, the NS2 models are not compatible to thus of NS3 and including them into a NS3 simulation is not possible [5].

- **J-Sim** has been developed in Java. It offers a framework for WSN simulation built on ACA, INET and Wireless protocol stack. It encompasses the following components:

1. Sink, Sensor and target Nodes.
2. Sensor Channels and Wireless Communication channels
3. Physical media such as Seismic channels, mobility models and power models.

J-sim implements three WSNs protocols: Localization, Geographic Routing and Directed Diffusion. The design of J-sim makes difficult adding new protocols or node components to users because it was not originally dedicated for WSNs. Therefore, the execution time is much greater than thus of NS-2 [6].

As discussed below, a large number of network simulators are available. But, they present several drawbacks in term of network deployment, data routing and performance metrics. First, in some simulators the user must manipulate command lines to plan its network and define its architecture. Indeed, He must resort to deploy sensor nodes one by one which is painful for the large scale networks. On the other hand, the architecture of a mobile network is likely to change over time since nodes can change their positions. This may not be applied because the simulator takes as input a topology file describing the network architecture which remains the same throughout the simulation. In terms of scalability, several simulators support a limited number of simultaneous node i.e. small networks and cannot simulate a large network. Second, for data routing, the majority of simulators have low efficiency in term of extensibility. In fact, they present a difficulty to add new protocols or node components within this simulator. It is also very difficult to visualize all the exchanges between sensors composing the network. Finally, among the existing simulators cited below, there isn't a complete simulator and they fail to model all the required performance metrics needed by a designer to validate its data routing protocol in a mobile WSN.

As conclusion, practical issues such as mobility modeling , energy modeling, loss packets rate modeling, sensors network specific models, large scale network support, real time simulation, real system implementation, extensibility to add new protocols, graphical display to see simulation parameters,... are important way to be found in a developed simulator.

3 Proposed simulator: Mobi-Sim

In its most general sense, the term "system" refers to a set of real or abstract units that interact to serve a common purpose. Modeling is used to create models for each element in the system environment in order to interpret an experiment as well as to predict how the system evolves and how it behaves in a given situation. This modeling serves to validate an operation by simulation and to refine the specifications [4]. Indeed, modeling and simulation have seen an unprecedented progress in various sectors of industrial engineering for the design of new systems in order to study their characteristics as widely as possible, which increasingly encourages methods for the development of modeling and simulation tools.

Indeed, according to the existing simulators limitations, presented in the previous section, mainly because they don't support the presence of mobile nodes in a network, do not model all the performance metrics intended by a designer to validate his new approach, present a difficulty for users to implement their new protocols, do not offer real-time simulation and the simulation in others does not model a real sensor... So, to accelerate the design process, a complete-fast and powerful simulation environment that meets all these requirements may be essential for the designers.

In this context, we are interested to "monitoring systems" based on a mobile Wireless Sensors Network. The main idea of our Mobi-Sim simulator is to emulate a sensor node on a laptop. Indeed, a laptop will react (with the components that surround it in its environment) as a sensor. The proposed testbed, launching a simulation, is shown in Figure 1.



Figure 1: Proposed testbed

Our simulation system is able to get closer to the reality of things, to reproduce the behavior of the elements which belong the network in real time, to simulate a network containing fixed nodes only, to simulate a network containing mobile nodes and to simulate a hybrid network (consisting of fixed & mobile nodes). It is dedicated for a mobile wireless sensors network to deploy a large number of laptops as mobile nodes. Therefore, it has an efficient extensibility to easily integrate new routing protocols and it presents a generic platform to implement any routing protocol. In fact, a laptop including Mobi-Sim will replace a sensor node that can executes any existing algorithm in its memory. For the performance metrics, Mobi-Sim presents

a set of performance criteria for a routing protocol namely the energy consumption, the delivery packets rate, nodes connectivity and packets loss rate. The Mobi-Sim characteristics are detailed in the next section.

3.1 Mobi-Sim development environment

Mobi-Sim template is a complete working project emulating a sensor node behavior in a WSN on a laptop. This sensor may be a stationary node operating either in a fixed network or in a hybrid network (containing both fixed and mobile sensors); as well as it may be a mobile node communicating by wireless means. This application is developed with an ergonomic interface that help users, specially for novices, to simulate and validate their proposed approaches. In fact, to provide researchers to add their own proposed protocols (by writing them and implementing them into the Mobi-Sim template), this template has tools that generate empty modules in which the eventual protocol code should be integrated. Mobi-Sim is programmed using the Microsofts .NET framework. It is developed in the language C#.Net providing a graphical user interface (GUI) for users to interact with the different tab of this application in online mode. This GUI allows the deployment, the supervision, the control, the managing and the tracking of the different sensor nodes automatically in the network based on their ability in the network to communicate together.

3.2 Operating mode

The developed Mobi-sim operating mode includes the node deployment, neighbor discovery, periodic data exchange and energy consumption supervision. Node deployment may be done manually by users if it is a fixed node (static deployment) or randomly if it is mobile node (random deployment). In a static deployment, the user can define for each fixed sensor node its eventual neighbors and define its level in the network topology as shown in Figure 2. Then, it should start to exchange data with its neighbors according to the defined protocol.

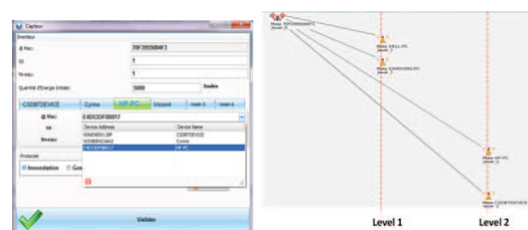


Figure 2: Static WSN deployment

If sensor nodes are mobile, they should be deployed using a random deployment as shown in Figure 3. The mobile sensor should discover automatically its neighbors with which it can communicate. The node discovery algorithm is based on Hello messages. Each node is defined by its MAC address (@MAC of the laptop implementing the node).

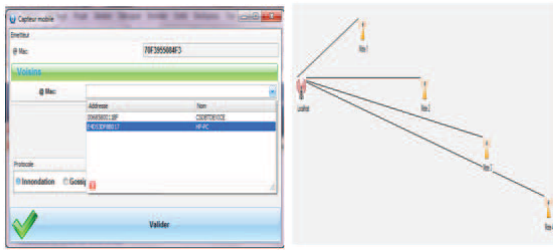


Figure 3: Mobile WSN architecture

The implemented functionalities in Mobi-Sim are shown in Figure 4.

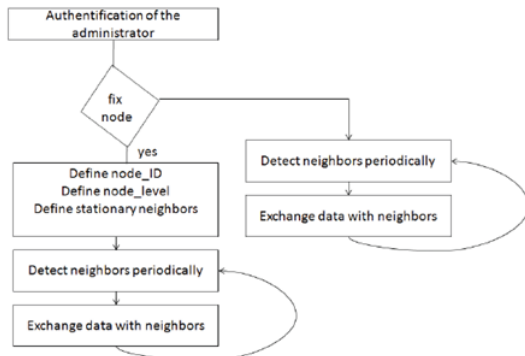


Figure 4: Mobi-Sim operating mode

3.3 Energy Model

The basic entity of a network topology is the sensor node. In Mobi-Sim, a laptop is represented as a node. In this simulator, the power component serves for power management. The consumed energy by a sensor node is essentially due to the following operations: data capture, processing and communication. The radio module is the component of the sensor node which consumes the most energy (about 94%) because it is responsible to ensure the communication between the nodes [17-18]. There are four states in the radio components (transmitter and receiver): active, receive, transmit and sleep.

The amount of energy required for both sending and receiving data from node j to another node, computed as in [19], are given by:

$$E_{Tx}(j, k, d) = E_{Trans} \times j \times k \times d^\alpha \quad (1)$$

$$E_{Rx}(j, k) = E_{Receive} \times j \times k \quad (2)$$

where E_{Trans} is the energy dissipated to transmit k bit along a distance d which separate the sensor node j to another sensor node. While $E_{Receive}$ is the amount of energy consumed by a sensor node to receive k -bits message. α is the path loss exponent due to wireless channel transmission [19-20]. In Mobi-Sim, these parameters are left to be set by users. Using this radio model, the transmission cost and receiving cost for a k -bit message with a distance d are: - Transmission:

$$E_{Tx}(k, d) = E_{elec} \times k + E_{amp} \times k \times d^2 \quad (3)$$

- Receiving:

$$E_{Rx}(k, d) = E_{elec} \times k \quad (4)$$

where: E_{elec} is the radio electronics dissipation, E_{amp} is the amplifier dissipation and d is the distance between transmitter and receiver. Mobi-Sim implements two type of sensor nodes: MICA2 and Telosb. A Simple_Power component has been implemented these four states. The power component response to the control signal from networking components. It can switch from one state to another depending on its operating mode. In the proposed energy framework, the parameters modeling such as initial energy are defined as attributes by user. Figure 5 shows a description of the various modules included in our framework. We implement the EnergySource and DeviceEnergyModel classes in this framework to represent energy sources and device energy models. Device_energy_model_container defines sensor nodes type (MICA2 or Telosb). Mobi-Sim provide users to add other types and define their attributes. In each node, device_energy_models consume energy from the energy_source. The energy source give notice for device_energy_models when it is completely drained. The proposed Mobi-Sim simulator energy framework provides functions that allow external simulation modules to send requests to the node's Device_energy_models. For example, a routing protocol may demand information about the battery level by querying the energy source on the node.

4 Mobi-Sim: Simulation & Evaluation

In this paper, we add other hierarchical protocols into our Mobi-Sim simulator: the LEACH mobile sink protocol and LEACH-M. In fact, hierarchical routing protocols have proved an efficient energy [21-26]. In the Hierarchical protocols, sensor nodes are dynamically organized into partitions called clusters. For each cluster, a cluster head (CH) is elected. The cluster heads have to collect and aggregate the data; thus reducing exchanged data packets and saving energy [27]. Instead of transmitting data directly towards the sink through long distances, sensor nodes have to transmit their data to the cluster to optimize energy consumption [28]. The implemented protocol treats the two aspects of mobility in a network: LEACH-M includes mobile nodes when LEACH sink-mobile include mobile sink.

Thus, we simulated them according to the previous cited criteria in [4] to show our simulator efficiency face to other popular protocols. We have also add another performance criterion which is sensor nodes connectivity. Then, we present another approach for data routing in a mobile WSN based on clustering that we simulate by Mobi-Sim.

4.1 LEACH-M Evaluation

LEACH-M is a protocol based on a dynamic network partitioning into a set of clusters. The protocol implementation involves three operating phases: a phase

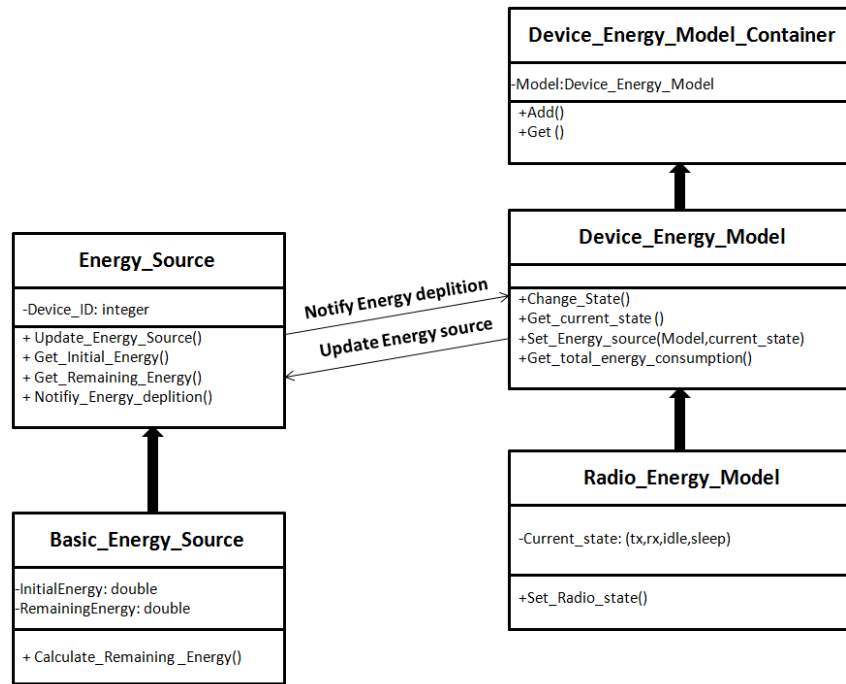


Figure 5: Class diagram of the proposed Mobi-Sim energy framework

of announcement and clusters creation, a scheduling phase and a transmission phase [28].

- *Announcement phase and clusters creation:* During this phase, the base station announces a new round. The different nodes self-elect to be CHs according to an election probability and a pre-determined percentage of CHs selection in advance [5% to 15%]. This election can be described as follows:
if $n \in G$:

$$\frac{P}{1 - P \times (r \bmod \frac{1}{P})} \quad (5)$$

Where P and r denote respectively the percentage of the nodes wanting to be CH and the current round, G represents the set of nodes which have not be CHs during the last $\frac{1}{P}$ iterations. n is a random value generated by a node in the range [0,1]. If n is less than a threshold T (n), the node is CH candidate and then should informs its neighbors about its election. Thus, an announcement containing its identifier as CH is broadcast via a CSMA/CA MAC protocol to avoid probable collisions and interferences between the adjacent CHs. The nodes decide to join the nearest CH. In case of equality, the choice of belonging to a CH is made randomly. This is done by sending a join packet by the sensor node to the chosen CH, the latter sends an acknowledgment message for confirmation.

- *Scheduling phase:* Once the clusters are formed, each CH moves from a simple member node

(MN) to a coordinator for information transmission within its group. Based on the task scheduling method, it implements the TDMA protocol and assigns to each of its MNs a slot time during which the node can communicate its information. All of these time intervals constitute a frame which duration differs depending on the number of nodes in the cluster.

- *Transmission phase:* It is a phase for gathering information. Based on the TDMA method for accessing the MAC layer, the MNs communicate their data to the corresponding CHs for a pre-defined period of time. The collected data at each CH are aggregated and then transmitted directly to the base station. Outside the allowed time for transmission, each node has the ability to standby to save its resources. In this protocol, the data processing at each cluster is done locally and each CH coordinate the exchanges with the other MNs. The network has the capacity to self-reorganize during the CHs election phase. Each node has the possibility to be elected CH and inversely, each CH can become a simple MN that can belong to a cluster. The CH election is based on the residual energy. It also performs data aggregation from MNs. Unlike the MNs, the CHs are in permanent activity since they communicate either with all the nodes belonging to their clusters or they exchange data with the base station.

The LEACH-M routing protocol make the following assumptions:

- The network uses N homogeneous nodes deployed

randomly and are able to communicate with The base station in mono-hop.

- In each cluster, except for the initialization phase, the new CH election is carried out by the CH of the previous round according to its residual energy.

Initially, the base station broadcasts a message to determine the number of clusters as well as the probabilities 'P' associated to the different nodes wishing to become CHs. In LEACH-M, the initial number of CHs corresponds to 10% of the nodes total number within a network. Each node receiving a message from the base station must rebroadcast it to its neighbors. It must generate a random number nb_a that will be compared to the probability P. If the probability P is less than nb_a then the node is a CH candidate, otherwise it waits new CH notifications from other CHs.

Once elected as CH, the node broadcasts its new state to all its neighbors by sending an advertisement message 'ADV' containing its identifier ID. Other nodes select their clusters based on their positions. Each MN must inform and join a CH in their coverage. Based on the scheduling method, the protocol implements the medium accessing method TDMA by assigning for each MN a time slot for communication. After having received, processed and transmitted its data, each node can pass to the Idle mode in order to optimize its energy. The different CHs perform data aggregation, data fusion and then transmit it to the base station. Then, they proceed, depending on the residual energy of each node, to the new CHs election. The old CHs inform the new CHs of their new state and transmit them the list of identifiers of all the MNs. In the following, we present and analyze the simulation results and experiments obtained during the implementation of this protocol on two simulators: Mobi-Sim and TOSSIM.

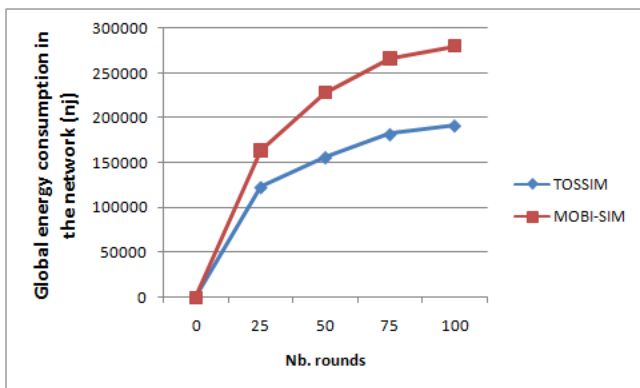


Figure 6: Average consumed energy in the network by LEACH-M

As illustrated in figure 6, the results of the LEACH-M implementation show that the energy consumed by sensor nodes is greater in Mobi-Sim than in TOSSIM. The obtained result is due to sensors nodes mobility which can change their positions over the time which make their supervision a difficult task for TOSSIM. In fact, unlike Mobi-Sim, TOSSIM cannot reproduce mobility behaviors and then mobile sensor nodes (which change their initial position) are not able to detect

and to send their data packets toward other sensors. So, TOSSIM loses the control at the moving nodes. These nodes will behave as disconnected nodes whose neighbors could not receive their data packets which will influence into the total rate of the energy consumption in the network. In against, Mobi-Sim could model and reproduce the behavior of mobile nodes as well as could supervise the exchanged data packets between mobile sensors whatever their position as long as they are in the sensing field. Since sensor nodes are able to be supervised while moving and they still able to communicate together, the exchanged data packets rate would be higher for Mobi-Sim (see figure 7) as well as the connectivity rate i.e. the number of active nodes (see figure8). Therefore, if the exchanged data packets rate increases, energy consumption will also be very important (because radio communication module dissipates the major amount of energy within a sensor node).

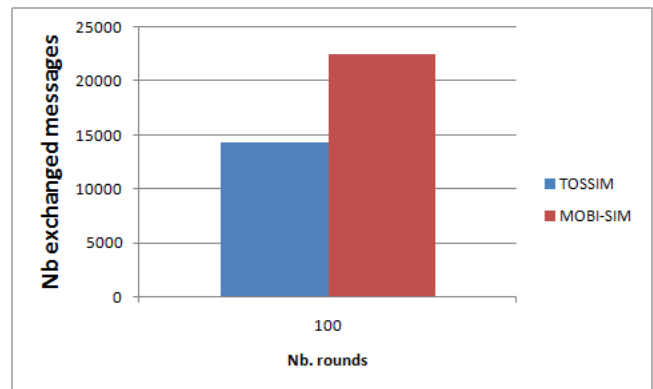


Figure 7: Exchanged messages number by LEACH-M

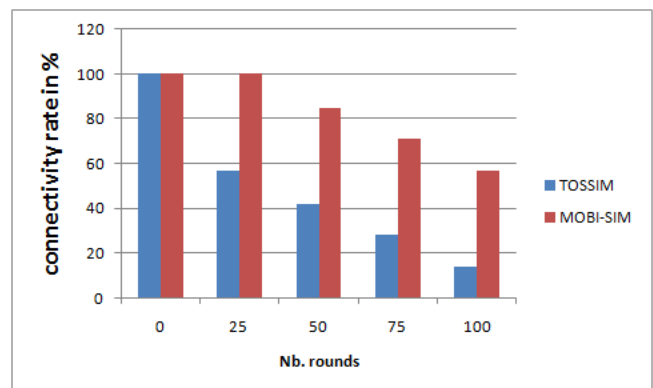


Figure 8: Connectivity rate by the LEACH-M in both simulators: Mobi-Sim and TOSSIM

The next performance criterion is the lost packet rate. The packets lost number is the difference between the number of the generated packets from sensor nodes within a network and the number of the received one at the sink. Figure 9 shows that the loss data packets rate is higher in TOSSIM than Mobi-Sim. This is mainly due to sensors mobility. In fact, when leaving an area to enter in another, TOSSIM consider this sensor node as died node which cannot deliver its data packets to its neighbors.

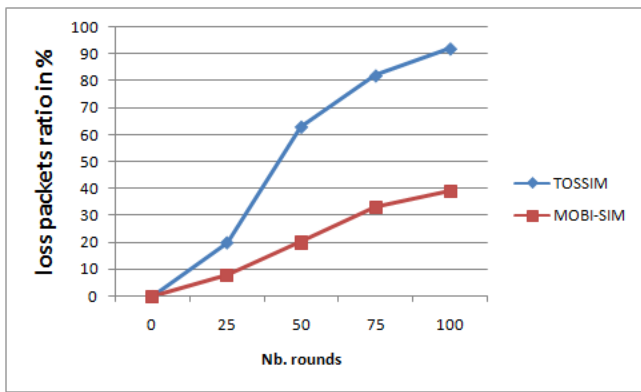


Figure 9: Loss packets rate by LEACH-M

Mobi-Sim presents better performances on the popular LEACH-M protocol and can model well the mobile sensor nodes behavior than TOSSIM in a mobile WSN.

4.2 LEACH Sink-Mobile Evaluation

The LEACH sink-mobile communication architecture consists to form clusters and to elect for each cluster a cluster header as a gateway to the base station. CHs are randomly elected (5% of the total nodes number) based on a probability function according to a specific election algorithm. Nodes member transmit their data packets to their CH. Then, each CH receiving data aggregates and transmits it to the BS directly or via other CHs. For the topological model, the BS is mobile and can move from an area to another one. LEACH Sink-Mobile is based on the following points:

- Sensor nodes are fixed,
- More than one mobile base stations,
- Each elected CH, broadcasts a message to other members in its cluster. MNs have to join this CH in order to send it their collected data,
- If a sensor node fail to recover the new CH ID, it waits this ID from the old CH after sending it its data,
- If the CH fails to find an itinerary to send data to the BS, it tries to join another CH in another cluster. Otherwise, it randomly selects a sensor node in the border to ask for a path to the BS [27].

The simulated result in Figure 10 exposed the considerable variation in the network lifetime for the LEACH sink-mobile protocol on two simulators: Mobi-Sim and TOSSIM. The energy consumption in the WSN using Mobi-Sim is greater because mobile sinks stay active as a whole and can be followed even if they move. This was not the case in TOSSIM which loses the control on mobile entities when they change their position.

The main metric in a routing protocol is its ability to find itineraries to the base station in order to deliver collected data from sensor nodes to it. On failure sensor nodes try to resolve this problem by retransmission. But, because TOSSIM don't support the presence of mobile entities in the network, any moving node will be considered as died node. For thus, sensor nodes could not locate the sink and the sink could

not communicate with other nodes. So, this is why the number of exchanged messages is less in TOSSIM than Mobi-Sim as shown in figure 11.

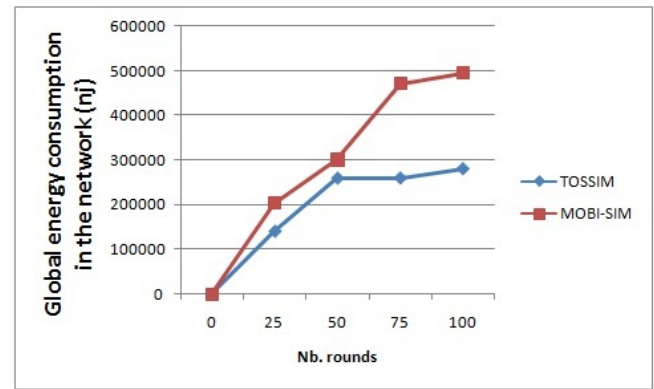


Figure 10: Energy consumption by LEACH sink-mobile

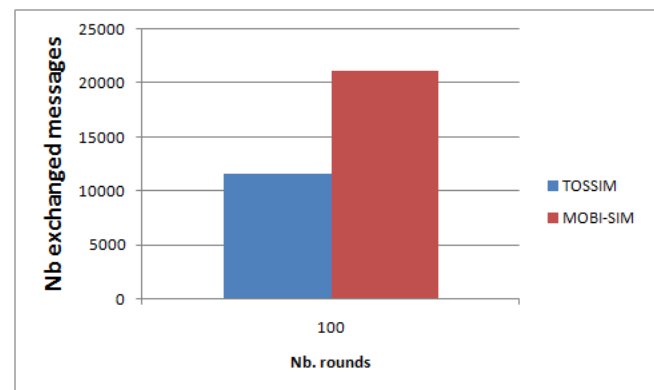


Figure 11: Exchanged packets number with LEACH sink-mobile

Therefore, since the mobile sink is considered as died node when changing its position, sensor nodes try to find it by retransmission and then they are likely to deplete their energy resource which may decrease the connectivity rate over the time and increase data loss packets. Inversely, Mobi-Sim supports the presence of mobile nodes and could follow it even if it is moving. This is explained in Figure 12 which show more connectivity rate respectively in Figure 13 which show less loss data packets rate in Mobi-Sim than TOSSIM.

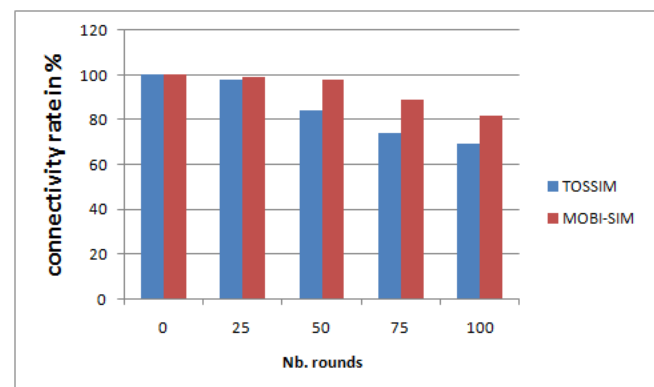


Figure 12: Connectivity rate by LEACH sink-mobile

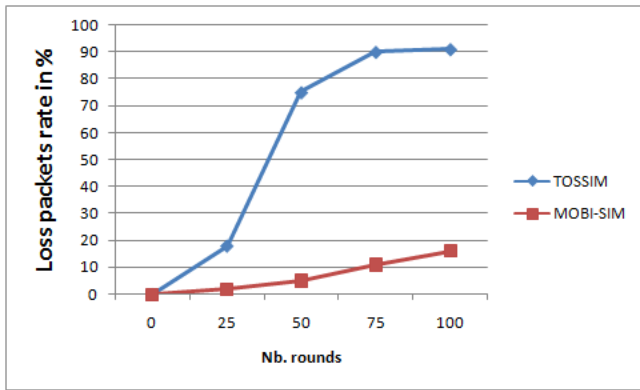


Figure 13: Loss packets rate in LEACH sink-mobile

4.3 Proposed Approach Evaluation

The proposed hierarchical routing protocol is a protocol based on a dynamic partitioning of the network into a set of clusters. Our goal is to ensure optimal operation in the network by minimizing the energy consumption and the delays when routing the information. This protocol is dedicated to operate with mobile sensors network and to support the frequent topology change.

To implement the proposed routing protocol, we make the following assumptions:

- The network uses N homogeneous nodes deployed randomly.
- The base station is fixed.
- There are few fixed nodes as reference points and a large number of mobile sensor nodes.
- In each cluster, the new CH election is carried out by the reference points periodically according to the energy for which the residual energy of each node within the same cluster is maximum and to the mobility for which the CH should have a low mobility rate.

This protocol implementation involves three phases: Clusters formation, Cluster heads election and data transmission.

- *Cluster formation:* During this phase, the reference points announce a new round to start the organization process. The different reference points broadcast an announcement to launch the clustering process. According to their adjacency, sensor nodes are organized into clusters.
- *CH election phase:* Once the clusters are formed, each reference point elects a coordination center role for each group. In each group, the transmission of information should be from member nodes toward the CH. This CH is elected based on energy criteria i.e if it has the maximum amount of energy and based on mobility if it has a low mobility rate compared to other members within a cluster.

- *Transmission phase:* The CH collects and aggregates data in order to transmit it to its corresponding anchor. Then, the anchor transmits its data packets directly to the sink or via other anchors. In this protocol, data processing in each cluster is done locally and the role of each CH is to coordinate the exchanges with other members. Each node has the possibility to be elected as CH and inversely, each CH can become a simple member that can belong to a cluster.

The operating mode of a sensor node in our proposed routing protocol is shown in Figure 14.

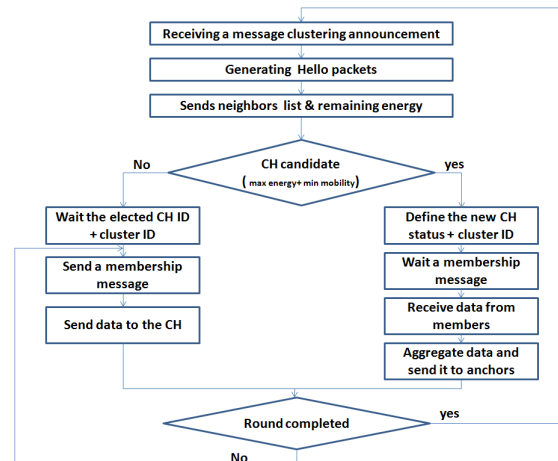


Figure 14: proposed routing protocol implementation

The proposed approach is the third protocol which is simulated in both Mobi-Sim and TOSSIM simulators. Like the previous results, Figure 15 show that the number of the exchanged packets is higher in Mobi-Sim than TOSSIM because unlike TOSSIM it can model and reproduce the behavior of mobile nodes.

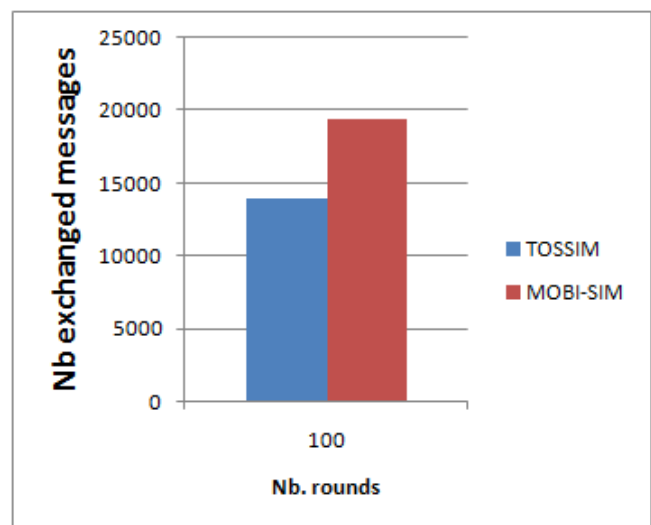


Figure 15: Exchanged packets number of the proposed approach

Figure 16 shows the average of consumed energy in the network over time. Since the number of exchanged packets in Mobi-Sim is greater than TOSSIM, the consumed energy is also higher for this simulator.

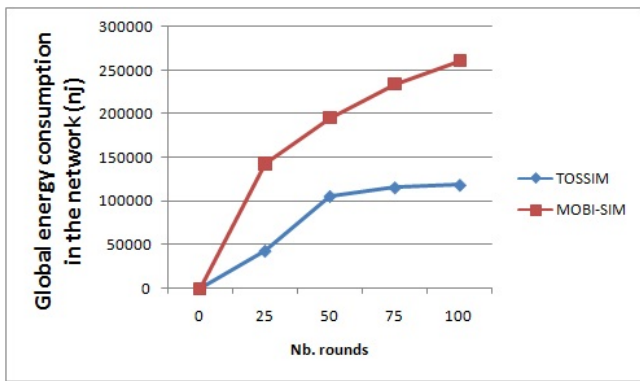


Figure 16: Energy consumption of the proposed approach

Like other protocols, TOSSIM cannot follow mobile nodes running a routing method dedicated for mobile WSN. In fact, the obtained results in Figure 17 show that the loss packets rate is greater in TOSSIM than Mobi-Sim. As TOSSIM loses the control on mobile nodes, the number of non-receiving nodes is increased. Then, the lost packets rate is increased in TOSSIM than Mobi-Sim.

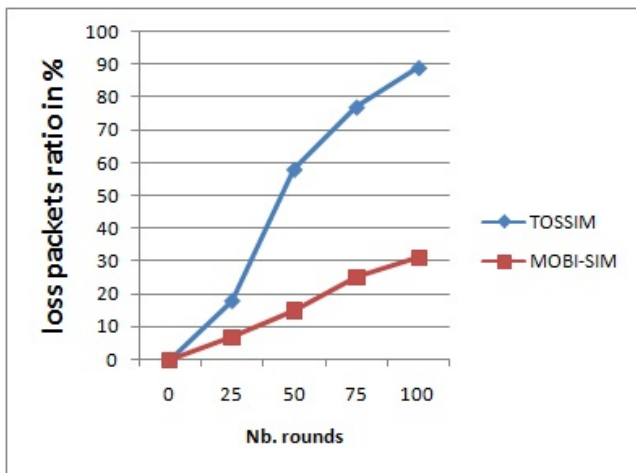


Figure 17: Loss packets rate of the proposed approach

The same for connectivity rate, Mobi-Sim supports the presence of mobile nodes and could track it even if it change its initial position. Thus, Figure 18 shows more connectivity rate in Mobi-Sim than TOSSIM. This justifies the previous results.

Therefore, to evaluate the performance of our proposed approach based on clustering, we simulate it and LEACH-M together on Mobi-Sim. Figure 19 shows the global energy consumption by sensor nodes within a mobile WSN with respect to different rounds. The obtained results show that LEACH-M do not provide an efficient energy consumption. The reasons behind this is that LEACH-M is a centralized protocol in which the base station is responsible for clusters formation, CH election and sensor nodes organization. However, our proposed protocol is distributed i.e. sensors nodes communicate together to perform sensors nodes organization and clusters formation.

Therefore, our approach which is multi-hop routing method, LEACH-M is a mono-hop protocol and CHs should communicate their data packets via long distances. So, it is evident from Figure 19 that there is a significant difference in the energy consumption between LEACH-M and our proposed protocol in the network, which directly impacts on the performance of the network or network lifetime.

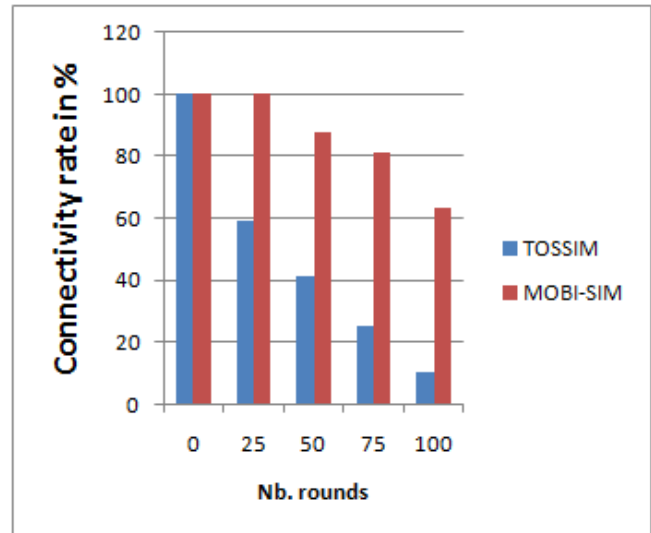


Figure 18: Connectivity rate of the proposed approach

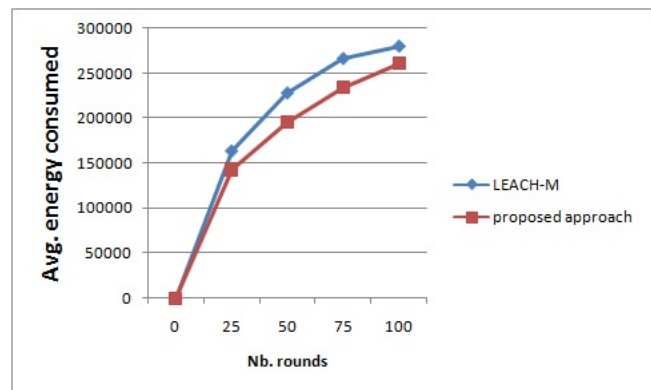


Figure 19: Sensors nodes energy consumption of our proposed protocol compared to LEACH-M

Figure 20 shows the loss packets rate. The obtained results show that the proposed protocol outperforms LEACH-M in terms of data delivery ratio. This is mainly due to the efficient CH election method which is based on the combination of two criteria: remaining energy and low mobility. In fact, sensor node with low mobility rate and maximal amount of energy in its battery is elected as CH. But, LEACH-M is elected initially basing on the remaining energy and on a generated random number compared to a probability threshold. So, data packets in our approach are more likely to achieve the base station than in LEACH-M.

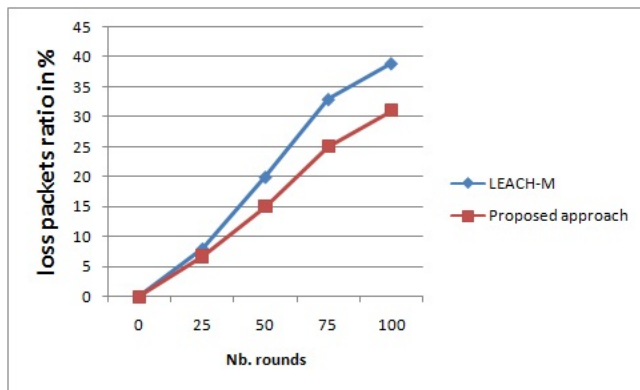


Figure 20: Loss packets rate of the proposed approach compared to LEACH-M

Figure 21 shows the connectivity rate between sensor nodes over time. The obtained results show that the number of active nodes over time is higher in our proposed approach than LEACH-M. This is behind many reasons: First, LEACH-M is based on negotiation i.e. when the CH fails to recover sent data by a sensor node in its time slot, it sends a request for two data frames then it considers it as mobile node. So, an extra traffic will be added which give chance for energy depletion which may quicker disable sensor nodes. But, in the proposed approach, the elected CH broadcasts an announcement message to inform neighbors by its state and then waits for joining message and data packets. Second, CH in the proposed protocol is elected if it has a low mobility. This offers a more efficient method.

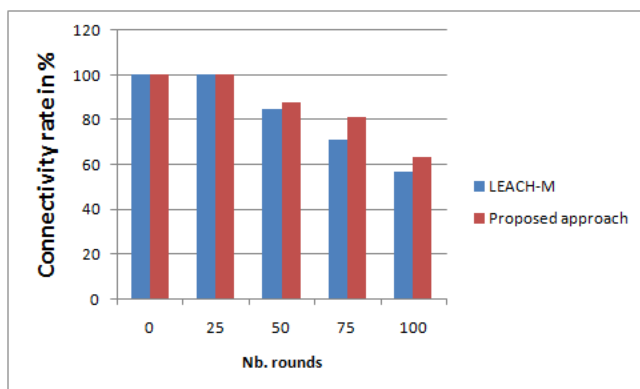


Figure 21: Connectivity rate of the proposed approach compared to LEACH-M

By all the performed simulations discussed above, the proposed simulator Mobi-Sim outperforms TOSSIM to support and to well reproduce the mobile sensor nodes behavior. Therefore, it efficiently model the loss data packets, the connectivity, the energy consumption and the exchanged data packets in real time. This simulator has also shown its efficiency when we implemented in previous work the flooding protocol as well two other popular routing protocols: LEACH-M and LEACH sink-mobile. Therefore, the proposed routing protocol based on clustering has demonstrated its performance compared to LEACH-M in terms of network lifetime, energy consumption, loss packets rate and sensor nodes connectivity.

5 Conclusion

The main objective of this paper is to provide a new simulation framework that can implement different WSN routing protocols to deliver the most accurate result. We adopt a model for mobile WSN that is able to reproduce mobile sensors behavior by implementing this sensor on a laptop. It is offered to researchers to evaluate the performance of their proposed approach before any implementation in a real system. We tested our model on two different popular routing protocols for better understanding. Simulation results have demonstrated that Mobi-Sim is suitable for mobile WSN and is open to add new approaches. Therefore, we proposed and simulated on Mobi-Sim a new routing protocol based on clustering. This proposition has demonstrated its efficiency in term of energy consumption and connectivity between sensor node as well as it demonstrated its robustness against loss packets.

Programming model is a huge step to deploy more applications in WSNs domain. There are currently many research works that addresses several parameters related to the sensors, such as: their positioning, energy, processing and data routing. We plan also to propose a routing protocol that supports nodes mobility and providing their localizations.

Acknowledgment This work has been accomplished at WIMCS-Team research, ENETCOM; Sfax University. Part of this work has been supported by MESRSTIC Scientific Research Group- Tunisia.

References

1. Pr. M. M. Abdellaoui, "Multitasks-Generic-Intelligent-Efficiency-Secure WSNs and their Applications", LAMBERT Academic Publishing (LAP), 2017, Part 2: Intelligent Wireless Sensors Network, pp. 108-141.
2. M. Mezghani, R. Gatgout, G. Ellouze, A. Grati, I. Bouabidi, M. Abdellaoui, "Multitasks generic platform via WSN", International Journal of Distributed and Parallel Systems (IJDPS), vol. 2, No. 4, July 2011, pp. 54-67.
3. O. Mezghani, M. Abdellaoui, "An Efficient Localization Method For Mobile Nodes Tracking in Wireless Sensors Network", IEEE International Conference on Communications, May 21-25, 2017, Paris. (submitted at publication).
4. O. Mezghani, M. Abdellaoui, "Mobi-sim: A Simulation Environment for Mobile Wireless Sensors Network", 3rd International Conference on Control-Decision and Information Technologies (CODIT2016), April 6- 8, 2016, Malte.

5. S. P. Singh, S.C. Sharma, "A Survey on Cluster Based Routing Protocols in Wireless Sensor Networks", International Conference on Advanced Computing Technologies and Applications (ICACTA), March 26-27, 2015, Mumbai, India.
6. G. Leelavathi, K. Shaila, K. R. Venugopal, L. M. Patnaik, "Design Issues On Software Aspects And Simulation Tools For Wireless Sensor Networks", International Journal of Network Security & Its Applications (IJNSA), vol.5, No.2, March 2013, pp. 47-64.
7. D. Navarroa, W. Dua, F. Mিয়েvillea, L. Carrela, "Hardware and Software System-Level Simulator for Wireless Sensor Networks", International Journal of Procedia Engineering, vol. 5, 2010, pp. 228-231.
DOI:10.1016/j.proeng.2010.09.089.
8. W. Du, F. Mিয়েville, D. Navarro, I. O. Connor, "IDEA1: A validated SystemC-based system-level design and simulation environment for wireless sensor networks", EURASIP Journal on Wireless Communications and Networking, No. 1, 2011, pp. 1-20.
DOI: 10.1186/1687-1499-2011-143.
9. I. A. Qutaiba, A. Abdulmaowjod, M. M. Hussein, "Simulation & Performance Study of Wireless Sensor Network (WSN) Using MATLAB", Iraq Journal of Electrical and Electronic Engineering, vol. 7, No. 2, 2011, pp. 112-119.
10. K. Satish, S. J. Rane, D. Vishwakarma, "A Simulation Study of Behaviour of Wireless Motes With Reference To Parametric Variation", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 1, No. 2, August 2012, pp. 91-95.
11. B. Musznicki, P. Zwierzykowski, "Survey of Simulators for Wireless Sensor Networks", International Journal of Grid and Distributed Computing, vol. 5, No. 3, 2012, pp. 23-49.
12. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, "Tinyos: An operating system for sensor networks", Werner Weber, Jan M. Rabaey, Emile Aarts, editors, Ambient Intelligence, 2005, pp. 115-148.
13. P. Levis, N. Lee, M. Welsh, D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications", 1st international conference on Embedded networked sensor systems (SenSys 03), 2003, New York, NY, USA.
14. V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, M. Welsh, "Simulating the power consumption of large-scale sensor network applications", 2nd international conference on Embedded networked sensor systems (SenSys 04), November 3-5, 2004, New York, NY, USA.
15. H. Sundani, H. Li, V. K. Devabhaktuni, M. Alam, P. Bhattacharya, "Wireless Sensor Network Simulators. A Survey and Comparisons", International Journal Of Computer Networks (IJCN), vol. 2, No. 5, 2011, pp. 249-265.
16. T. Voigt, R. Sauter, V. Reynolds, "Towards Comparable Simulations of Cooperating Objects and Wireless Sensor Networks", 1st International Workshop on Performance Methodologies and Tools for Wireless Sensor Networks, October 23, 2009, Pisa, Italy.
17. G. Dimitriou, P. Kikiras, G. Stamoulis, I. Avaritiotis, "A tool for calculating energy consumption in wireless sensor networks", Advances in Informatic, 2005, pp. 611-621.
18. G. Mathur, P. Desnoyers, P. Chukiu, D. Ganesan, P. Shenoy, "Ultra-low power data storage for sensor networks", Journal ACM Transactions on Sensor Networks (TOSN), vol. 5, No. 4, 2009, pp. 96-101.
19. G. S. Kumar, V. Paul, K. P. Jacob, "Mobility Metric based LEACH-Mobile Protocol", 16th International Conference on Advanced Computing and Communications (ADCOM 2008), December 14-17, 2008.
20. A. Shahzad, B. Sardar, "An Intelligent Routing protocol for VANETs in City Environments", 2nd International Conference on Computer, Control and Communication (IC4), February 17-18, 2009, Pakistan.
21. K. Chan, H. PishroNik, F. Fekri, "Analysis of Hierarchical Algorithms for Wireless Sensor Network Routing Protocols", Wireless Communications and Networking Conference, IEEE, March 13-17, 2005.
22. X. Huang, H. Zhai, Y. Fang, "Lightweight Robust Routing in Mobile Wireless Sensor Networks", Military Communications Conference (MILCOM 2006), IEEE, October 23-25, 2006.
23. Y. Miao, H. Jason, L. Renato, "Mobility Resistant Clustering in Multi-Hop Wireless Networks", Journal of Networks, vol. 1, No. 1, May 2006, pp. 12-19.
24. S. Madani, D. Weber, S. Mahlknecht, "A Hierarchical Position Based Routing Protocol for Data Centric Wireless Sensor Networks", 6th IEEE International Conference on Industrial Informatics (INDIN 2008), July 13-16, 2008, Korea.

25. F. Tashtarian, A. Haghghat, M. Honary, H. Shokrzadeh, "A New Energy-Efficient Clustering Algorithm for Wireless Sensor Networks", 15th International Conference SoftCOMon Software, Telecommunications and Computer Networks, September 27-29, 2007, Portsmouth, UK.
26. X. Zou, B. Ramamurthy, S. Magliveras, "Routing techniques in wireless ad hoc networks classification and comparison", 6th world multiconference on systemics, cybernetics, and informatics (SCI 2002), July 14-18, 2002, Orlando, Florida, USA.
27. O. Mezghani, M. Abdellaoui, "Improving Network Lifetime with Mobile LEACH Protocol for Wireless Sensors Network", 15th International conference on Sciences and Techniques of Automatic control & computer engineering (STA2014), December 19-21, 2014, Hammamet, Tunisia.
28. G. Renugadevi, M. G. Sumithra, "An Analysis on LEACH-Mobile Protocol for Mobile Wireless Sensor Networks", International Journal of Computer Applications, vol. 6, No. 21, March 2013, pp. 38-42.