Visualization, Data Analyzing and Energy Usage Analysis in Wireless Sensor Network Based on TinyOs and PowerTossimZ

Mahmood Safaei, Abdul Samad Haji Ismail

Abstract—Design and implement thousand of sensor nodes for develop and test new algorithm in realistic is difficult and sometimes is impossible. Cost of time and cost of money cause for using simulations. Hence many kind of simulation are available such PowerTossim, NS2 and many more. These simulations are very helpful for reduce the costs. But some times lack and weak point in simulation can cause many difiiculty for user. SmartSim has been develop for solve many problems in PowerTossim. It has many useful features such as user graphical interface, compile the configuration file automatic, very fast report generator and so on. In this paper produce and develop some unique feature in plotting based on PowerTossim and SmartSim. This module can help to SmartSim to be unique all-in-one simulation for TinyOs and Tossim.

Keywords—SmartSim; TinyOs; Tossim; PowerTossim; Simulation

1 Introduction

WIRELESS communication is one of the best aids for wide variety of research in different elds like in automotive industry and health eld. Therefore sensors with small size and wireless communication are in high level of imortance.This has generated new studies about Wireless sensor Network (WSN). In WSN there is many facilities used such as hardware tools (sensors, motes) and software tools (OSs, Simulations) which makes WSN both useful and challenging. Thanks to this variety user has more choices in his study but difculties rise from constant and fast updates and upgrades of tools.

This paper introduces rst some general information based on Wireless Sensor Network and Graphical User Interfaces (GUI) based on different Operating Systems (OS). Next it demonstrates SmartSim, a TinyOS and Tossim simulation and an implementation of a new GUI with several unique abilities, excellent illustration of the topology and an easy interactive mode with the user. In the end there will be shown a modication of PowerTossimZ 3 for getting useful and easily understandable outcome from simulations.

2 Background And Related Work

2.1 OS

Operating System (OS) plays a crucial role in Wireless Sensor Networks due to the limitation of power and small size of hardware such as RAM and CPU in a sensor node. With these limitations in resources, a very heavy OS is not practical for sensors and some specially designed OSs are needed such as Contiki [1] and TinyOS. TinyOS is a light operating system based on event drive architecture. With TinyOS it is possible to control and access to radio communication, EEPROM and other hardware parts. TinyOS is written with a light version of C language that is called NesC. TinyOS supports Mica2 sensors, Epic, Mulle and Shimmer2 platforms [2], [3]. TinyOS also is a component base OS which means it is created by many independent components that communicate with each other in different levels with commands. After a command has been issued, it waits for signals. Another important feature of TinyOS is power management. It started with TinyOS version 1 with On/Off devices. TinyOS version 2.x uses completely different power method for operate calcul tion [4].

2.2 Simulations

Wireless Sensor Network (WSN) is a wireless network consisting of several sensors and created to test scientific algorithms which researchers need to test either with real device or in simulation environment. There are several simulation environments for WSN. For instance, Cooja is a simulation based on Contiki OS. Cooja has an interface that user can interact with in two major parts.The first one is the simulator and the second one is the core. The interface in simulator is called a simulation interface. In this simulation interface, it is possible to access to the memory. In the core interface user can access to a Contiki function.Sometimes it is necessary to access both, simulation interface and core interface; a good example

- *M.Safaei is with the School of Computer Science (Computer Network and Security), University Technology Malaysia*
- *A. S. Ismail is associated professor at the Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Skudai, Malaysia 81300. E-mail: abdsama@utm.my*

of this situation is radio transmitting or receiving data [1].

Tossim is another simulation environment for WSN that comes with TinyOS and it takes its ability from TinyOS architecture. Tossim offers an external communication platform that can monitor the communication between nodes, transition of packets and also injection of a new packet to the network. Tossim provides three kinds of network connectivity:

- Simple connectivity which means all sensor nodes are in one cell and establish a connection.
- Static connection where simulation creates the network graph at beginning of simulation and it never changes after that.
- Space connectivity where sensor nodes can move randomly in a square area. Random movement in space connectivity forces sensor nodes to change transmission range for keeping up the connection between nodes.

With implementation of a new TinyOS version some features of TinyOS cannot be supported by Tossim so an extension of Tossim has been created to solve some of its weaknesses, especially on power consumption. This extension called PowerTossim has very high efficiency inherited from Tossim. With this character it can simulate large topology of sensors network. One of the features of PowerTossim is the ability to simulate hardware components in sensor node such as CPU, radio and LEDs. PowerTossim uses a function called Power State for calculating power usage on each hardware component and this component can help calculating total power usage in each sensor.

2.3 Graphical Interface/Visualization

The technique that uses images, charts or animations for understanding better a message or improving communication with other things, is called visualization. Visualization or graphical interface makes it easier to understand things; for example in Wireless Sensor Network simulations export file, output or results of simulation which are mostly based on text files. Analyzing text file for getting the needed information is very difficult, especially when data has been gathered in large network topology of nodes. It is a big challenge for users so several GUI/visualization software has been generated and just like simulations they also have different versions anddesigns for different simulations.

A GUI software can have interaction or control on simulation environment. TinyViz is GUI based on Tossim and TinyOS version 1.x. TinyViz has some strong features such as plug-in design which gives users possibility to add customized plug-in suitable for their purpose. TinyViz is capable to interact with Tossim; TinyViz can start and stop simulation and user can also change some information from interface and again compile and start simulation with new configuration. This ability is due to the architecture on TinyViz. It uses the same event bus as Tossim and this factor not only gives ability to start and stop simulation but also ability to show the real simulation. TinyViz can show LEDs On/Off, radio transmission and also some report about sensor nodes. The challenges with TinyViz are that it is difficult to compile a new plug-in and also it does not have any version compatible with TinyOS version 2.x. Due to these challenges, there is currently no GUI available for PowerTossim because it works with TinyOS version 2.x [5], [6].

Visualization software or GUIs are not just for simulating environment. They can also be used in realistic environment or in an emulation mode. There are different variations of visualization software available since different sensor node models need different simulations. For example, Spyglass is software that is compatible with ESB 430/2 sensor board. This software works with actual devices which means the sensors are gathering data and sending it to the gateway and gateway sends the data to the visualization machine Spyglass has installed. Spyglass uses TCP/IP communication between gateway and visualization machine and due to it many different kind of media can be used such as wireless, GPRS or wired network. Spyglass uses plug-in architecture which gives its user the ability easily extent plug-ins or edit them. For designing purposes Spyglass uses Java2 standard edition and Java2D. Currently five plug-ins are available for Spyglass:

1)Temperature map plug-in gets temperature from sensors and draws temperature distribution by changing the background color where temperature is sensed.

2) Temperature plug-in uses same information that temperature map plug-in but shows only one temperature at the time.

3) Node painter shows a rough presentation of a sensor node.

4) Battery plug-in shows battery usage with information that sensor periodically sends to gateway.

5) Topology plug-in shows topology and connectivity between neighbouring sensors or nodes and also between them and gateway.

For these calculations sensors broadcast their information and to show the broadcast message Spyglass draws the line from source to all destinations [7].

Crossbow is a very famous sensor node that has a special OS and its very own simulation. Crossbow sensors have a specific GUI-emulator environment called Mote-VIEW. This software is built with Microsoft.net and it uses SQL server for saving data that is received from sensors in database. This option gives a very important ability to the software; when user wants to analyze data again, data is easily available. Mote-VIEW is designed on base of a layer design and it is split to four layers:

1) Data Access Abstraction Layer (Data layer).

2) Node Abstraction Layer (Node layer).

3) Conversion Abstraction Layer (Conversion layer).

4) Visualization Abstraction Layer (Visualization layer) [8].

All systems in the world have their pros and cons. TinyViz software cannot run based on Tossim's new version, PowerTossimZ 3, that is compatible with TinyOS version 2 and above. This weakness can cause many difficulties for user because there is no GUI software available for the newest version. On the other hand shortcomings there were not due to GUI rather than a lack of a report generator and an easy way to compile Tossim can increase the popularity and usage of TinyOS and Tossim.

3 SmartSim

In this and next sections will be highlighted the new software that is designed to solve the problem in new TinyOS and Tossim and improve ability of PowerTossimZ. Graphical Wireless Sensor Network simulation based on Tossim and TinyOS called SmartSim is a GUI based on Python programming for TinyOS 2.1.x and PowerTossimZ 3.1. SmartSim has much more characteristics compared with previous software. The rest of this paper introduces these characteristics.

3.1 SmartSim Architecture

SmartSim is written with Python, Pyqt4 and MatPlotLib. Python is a very powerful programming language and it is free to download and use. Python's ability is in combination of Java and C++ hence Python has a complete library of its own and a high functionality. Pyqt is a GUI toolkit for Python based on Qt. Pyqt binds Python with Design Graphical Interface based on Python. Pyqt is not the only one binding Python but also PySide, PyGTK, and wxPython are Python's GUI binders. Pyqt contains different modules based on Python. SmartSim has been used windowing design where different parts can work together but in completely separate windows with their own controllers. This design not only gives to user an ability to switch between windows without closing the current window but also will it give possibility to add expanded interface in future for some special purpose so that SmartSim can be split into three parts: 1) main window interface, 2) graphical simulation interface and 3) report generator interface. This independent window design can help to edit and add new features to the modules and interfaces without any effect on other modules.

3.2 SmartSim Features

Main Window Interface is the first window user interacts with while using SmartSim. This window has two main parts: 1) config.txt file information and 2) control buttons (Figure 1). In config.txt file part user can easily configure the text file that Tossim uses for simulation. This file has special information such as channel parameters, radio parameters and topology parameters. In SmartSim user needs just change variables in combobox or spinbox according to what he wants to simulate. The special parts

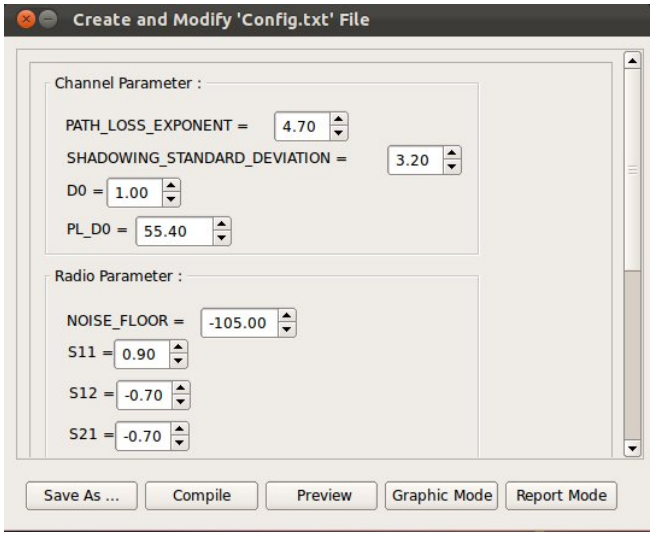

Fig. 1. Parameter Configuration Window Interface

in this window are the functional buttons. With Save As button user can save the config.txt file where ever he wants to save it in and have an easy access for future usage. One of the important buttons is Compile button. With pressing this button some particular commands will start to run. This specific button has an interaction with LinkLayerModel in TinyOS and it compiles the text file with a Java class and then gives two output files: 1) linkgain.out and 2) topology.out. The linkgain.out is used in another Python file for simulating the network with number of sensors that user has defined. The topology.out is used when Graphical Interface draws sensor nodes in the exact place user has defined. These two output files are used in different directories (Figure 2).

Graphical Simulation Interface is an analyzer for output file of PowerTossim. This interface has some unique characteristics that are available only in SmartSim (Figure 3). Graphic Window Interface is built of two different classes: 1) the class that analyzes the energy.txt file, the output of PowerTossim, and 2) the graphic objects. In SmartSim each sensor node is made of 16 different objects and they show sensor and events in more details. One sensor node includes three LEDs, antenna, CPU and battery gadget and all these objects have different characteristics that graphical interface shows. Other features of graphical interface are in control buttons. This interface can run simulation automatically and change the time of auto simulation from 0 second to 10 seconds. Simulation can also be used manually so that only one or group of 100 events(or whatever amount is programmed) happens after pressing Forward or Fast forward buttons. Other unique buttons in SmartSim are Backward and Fast Backward simulation buttons. With these features user can reverse the simulation one by one or again by group of for example 100 events. Forward and Backward buttons are unique for SmartSim and they have not been available in any other graphical simulations based on TinyOS and Tossim (Figure 4).

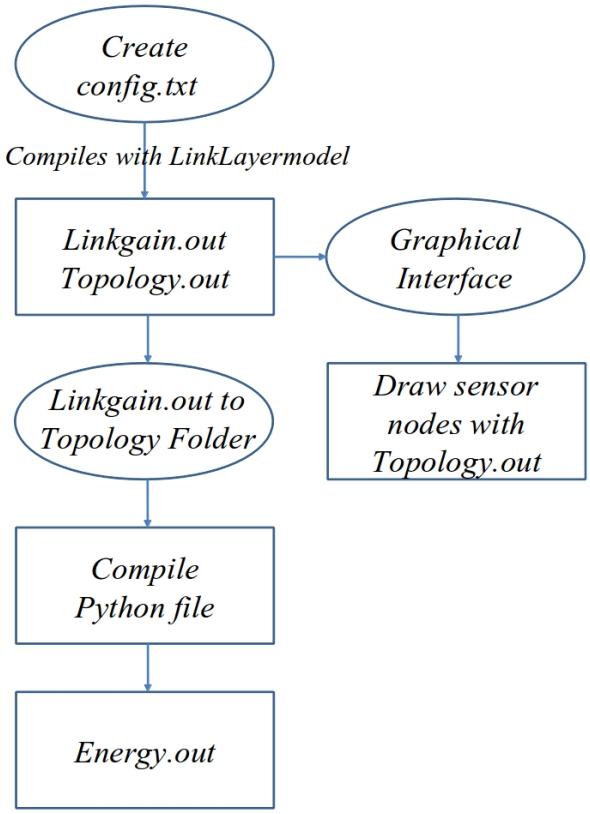

Fig. 2. Compilation Process Flowchart

Fig. 3. Graphical Simulation Interface

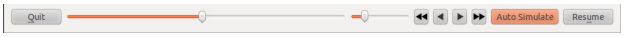

Fig. 4. Graphical Simulation Button Controller

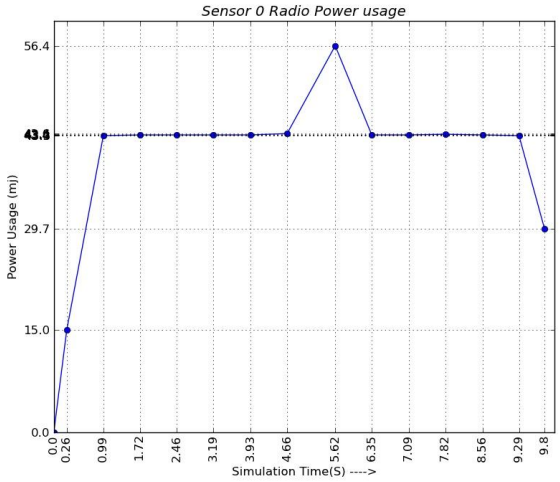

Fig. 5. Node 0 Radio Power Usage During Simulation

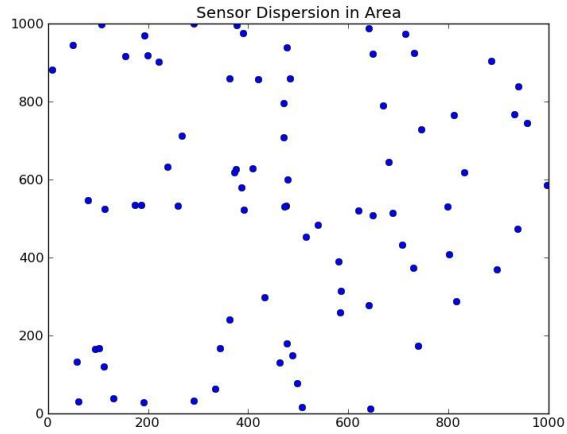

Fig. 6. Nodes Distribution Area

3.3 Report Generator Interface

SmartSim uses a very complete and useful plot designer. This ability is inherited from MatPlotLib, a powerful plot designer based on Python. SmartSim can illustrate different kind of reports based on energy in different parts of a sensor, such as radio and CPU, during simulation time (Figure 5) and also it can show information of all nodes based on specific part in topology for comparison (Figure 6). Also with SmartSim it is possible to analyze the difference between linear and nonlinear power usage in topology during certain period of time which is unique compared to all other software that has been design based on TinyOS and Tossim(Figure 7). In SmartSim user has capacity for getting results based on various kind of extension such as pdf, png and eps. Another special ability is that user can zoom or move on plot very easily for better analysis. In addition SmartSim can illustrate graph with details based on specific time or cumulative charts with more than 10 different kind of reports. based on simulation

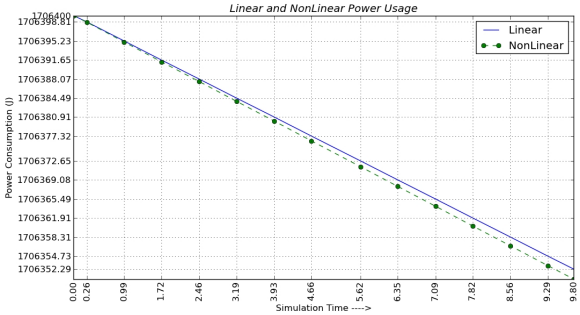

Fig. 7. Linear and Nonlinear Power Usage

4 Conclusion

As mentioned earlier WSN is a very common field for research and many OSs and simulations has been generated for it. This area is very critical because of limitations in hardware sources and node size. However implementing algorithms on real testbeds could be very costly. SmartSim has been generated for bringing a new GUI for TinyOS 2 and PowerTossimZ and also give new, very advanced features to user for creating and compiling the configuration file. It is also a very complete report generator that was not available in any GUI before. SmartSim is first and only all-in-one simulation based on TinyOS version 2 and PowerTossimZ 3 which presents topology design, algorithm troubleshooting and energy usage reports, all together.

References

[1] Fredrik Österlind, "A sensor network simulator for the Contiki OS," in *SICS Research Report*. Swedish Institute of Computer Science, 2006.

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *ACM SIGARCH Computer Architecture News*, vol. 28, no. 5, pp. 93–104, Dec. 2000.

[3] V. Shnayder, M. Hempstead, B.-r. Chen, G. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. Baltimore, Maryland, USA: ACM, 2004, pp. 188–200.

[4] E. Perla, A. O. Catháin, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "PowerTOSSIM z: Realistic Energy Modelling for Wireless Sensor Network Environments," in *Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks - PM2HW2N '08*. New York, New York, USA: ACM Press, 2008, pp. 35–42.

[5] B. Parbat, A. Dwivedi, and O. P. Vyas, "Data Visualization Tools for WSNs: A Glimpse," *International Journal of Computer Applications*, vol. 2, no. 1, pp. 14–20, May 2010.

[6] A. Kellner and K. Behrends, "Simulation Environments for Wireless Sensor Networks," no. June, 2010.

[7] C. Buschmann, D. Pfisterer, S. Fischer, and A. Kröller, "SpyGlass : A Wireless Sensor Network Visualizer," in *Special Issue on the Best of Sensys 2004 Work-in-Progress*, New York, NY, USA, 2004.

[8] M. Turon, "MOTE-VIEW: a sensor network monitoring and management tool," in *The Second IEEE Workshop on Embedded Networked Sensors*, 2005, pp. 11–18.