Encryption Techniques for Different Introducer’s Attack in Wireless Sensor Networks

Deepak Choudhary*

Electronics and Communication Department, ABES Engineering College Ghaziabad - 201009, Uttar Pradesh, India; engg_deepak@yahoo.com

Abstract

Wireless Sensor Network is described using the MARTE profile. In the UML/MARTE models Wireless Sensor Network is specified, defining the nodes that composed the network and characterizing the communication established among these nodes in the network. There are numerous ways to attack a Wireless Network. In this paper we study these techniques and propose alternative methods to simulate these attacks using a virtual simulator framework based on native HW/SW simulation methodology. Bearing in mind the emerging market in Wireless Sensor Network technology and its high vulnerability to attacks, it is important to provide new simulation techniques that allow the developer to obtain estimations in early stages of system design.

Keywords: Energy, Jamming, Performance Estimation, Simulation, Wireless Sensor Network, UML/MARTE

1. Introduction

In recent years, wireless communications have evolved rapidly, today being common technology in many applications in military, health, ecological, home and commercial-related areas. In order to cover many more potential applications and unique challenges, several advances in wireless communications have enabled the development of low-cost, low-power Wireless Sensor Networks (WSN). Commonly, this kind of network consists of a large number—from tens to hundreds, and even thousands—of low-cost, low-power resource-constrained multi-functioning sensor nodes, often operating in an unattended, hostile environment, with limited computational and sensing capabilities, to full-fill different application network, the latter has a number of additional vulnerabilities due to the shared medium access, unsafe and unprotected communication channel, broadcast transmission media, deployment in hostile environments, limited resources and bandwidth, power management, spectrum use, system complexity and health concerns. The security level required might vary from one application to another according to the importance of the information that is being exchanged.

One of the biggest problems of these types of networks is their vulnerability to being attacked in a huge number of different ways. As mentioned above, these networks are often deployed in an insecure environment, where an attacker has direct access to the nodes of the network. For this reason, it is of great importance at the design phase to take into consideration the weaknesses of the sensor network. Furthermore, understanding the potential effects of each attack on a node or the entire network itself helps to identify and prevent other problematic vulnerabilities. Thus, understanding the consequences of an attack is of great value for redesigning the WSN.

Model-driven design methodologies are commonly being adopted to handle the design of large, complex systems. Nowadays, UML is being used to deal with the design of electronic, real-time, embedded systems. Nevertheless, UML lacks the specific semantics required to fully support specification, modelling and design of current electronic systems. In this context, the standard MARTE profile was developed in order to model and analyse real-time embedded systems, providing the

*Author for correspondence
Encryption Techniques for Different Introducer’s Attack in Wireless Sensor Networks

...necessary for some important reasons:

The aim of this work is to include an attack model in a software virtual simulator, which enables the effects of these attacks to be studied in a WSN. This paper proposes a methodology based on UML/MARTE models to specify a WSN. The UML/MARTE models capture the WSN structure and the most relevant data transmission characteristics in order to specify the WSN to simulate. From these models, the required information for WSN simulation can be extracted. More specifically, we focus on simulating Jamming attacks as these are one of the easiest ways to compromise the availability of a WSN. For this reason, we use a WSN simulator that estimates all the features that could affect the attacked nodes. The technique used in this paper extends the performance analysis tool explained in [4]. The paper has been structured as follows: Section 2 describes the motivation for developing the simulation tool. After that, considerations about energy constraints in WSN will be presented in Section 3. Section 4 describes the different techniques of jamming attacks. Section 5 explains the simulation technique used. The description of the jamming attack modelling will be presented in Section 6. Section 7 explains how the modelling is doing using UML/MARTE. Section 8 reports the experimental results and Section 9 concludes this paper.

2. State of the Art and Motivation

As has been already stated, the benefits of knowing information about the damage produced by an attacker in the early stages of the design enhance prevention mechanisms and improve network performance. This work is necessary for some important reasons:

- Developers need to know of the vulnerabilities of their application software, hardware platforms or network structures at an early stage in the development. Fast and accurate simulation can provide information to the developer that enables the modification of the algorithms or the architecture of the network in order to avoid or minimize attack effects.
- One of the critical constraints in the WSN is its battery life cycle. In this type of network, the death of a node may cause the isolation of a part of the network.

Most of the attacks try or produce a node workload increment in the node, and therefore an increase of the node power consumption. This leads shorten the life cycle of the node.

- The evaluation of the node software code against attacks is a critical security aspect that is currently difficult to estimate at early stages of the design process.

Current WSN simulator tools do not offer the possibility to simulate typical attacks that these networks can suffer. To reflect the state of art about WSN simulation frameworks, NS-2 [5] and OMNET++ [6] are discrete event network simulators. GloMoSim [7] is a simulation environment for wireless and wired network systems. Another framework, TOSSIM [8], is a bit-level discrete event simulator and emulator of TinyOS [9]. Avrora [10] provides a clock-cycle accurate execution of programs that are executed in the WSN node. As far as we know there is no previous work that provides real-time Operating System support, power and execution time estimation for WSNs. Power consumption estimation is one of the critical aspects of WSNs. As mentioned above, one of the main objectives of the attacks is to increase the consumption of the node. To make an accurate estimation of the attack effects, it is important to use a performance analysis framework that provides power consumption and execution time estimations.

3. Energy Constraints in WSN

The evolution of WSN has led to low-power and low-cost deployments. Most WSN nodes are inherently resource constrained due to the limited hardware capability, thereby influencing directly in the security grade to implement in the network. Thus, the design of security services in WSN must consider not only the node’s energy resources, but also its memory, computational capability, availability and, related to these, its security constraints.

Energy is the greatest constraint in WSN capabilities. Generally, sensor nodes are battery-powered devices, in which their useful life depends on their battery life. These nodes are often placed in hostile environments with difficult access after deployment. Therefore, the battery charge taken with them to the field must be conserved to extend both the individual node’s life and network availability. There are other factors that hinder recharge ability, such as the number of nodes deployed or their scattering within the network. When considering implementing a security mechanism within a sensor network, the impact on the sensor node’s available energy must be considered.
4. Jamming Attack

Jamming can be a huge problem for wireless networks. It works by denying service to authorized users when legitimate traffic is jammed by the introduction of noise in the environment. Currently, the jammer has different strategies to achieve their objective. Figure 1 shows a scheme of these strategies.

4.1. Noise Jamming

The jammer emits a random noise signal. Noise can be introduced at different frequencies of the band. The effects are not similar, but we have to consider that it is not necessary to occupy all the bandwidth to break the communication. This type of attack can be classified as BroadBand Jamming, Partial Band Jamming and Narrow-Band jamming:

- **BroadBand Jamming** (Figure 1b): Broad-Band Noise (BBN) introduces energy over the whole bandwidth of the frequency spectrum that the application uses. The problem of using this type of attack is its low level of jamming power. This power is divided over a wide part of the spectrum.

- **Partial Band Jamming** (Figure 1c and 1d): It introduces energy in a specific part of the spectrum. It only covers some channels. As we can observe in the Figure 1, we can introduce noise in continuous or in discontinuous channels.

- **NarrowBand Jamming** (Figure 1e): It introduces energy only in one channel. The efficiency of this type of attack depends on the knowledge of the application.

Figure 1. Principal jamming strategies scheme (a) Spectrum channels, (b) Broad Band noise, (c) Continuous partial band noise, (d) Discontinuous partial band noise, (e) Narrow Band noise.

5. Simulation Techniques

A WSN node is typically based on a System-On-Chip (SoC). Most of the WSN nodes are composed of typical components, such as processors, memories or caches, plus two important hardware components, the transceiver and the sensor.

5.1 HW/SW Co-simulation

The co-Simulation methodology described in this paper is based on the native simulation approach explained in 4. This approach consists in the execution of annotated software code in an environment that models the platform hardware details. The original environment supports the execution of the software application over the hardware model. As presented in 14, the methodology used allows the hardware model to be described using XML files. Interpreting these XML files, the simulator tool builds the system model and performs the simulation. Thanks to these XML files, the simulator can be easily integrated into a Design Space Exploration flow.

5.2 Wireless Network Simulator

The virtual simulator framework has some important features that enable accurate simulation of the attacks. The first feature is power consumption estimation, which is one of the most critical consequences of attacks. Another important feature of the simulator is that it executes the same software code that will be executed in the WSN network. This is important because most of the previous simulator frameworks obtain their network traffic from external functions, not from real traffic. Without real traffic information, it is not possible to perform accurate simulation of the firmware of the nodes and attack detection. Moreover, the simulation provides fast estimations. This is important because it allows the developer to detect potential problems in early stages of the design. This shortens time-to-market and reduces costs, two of the most important market requirements.

The proposed framework supports classical WSN RTOS such as FreeRTOS15. Moreover, it allows a node-level simulation. Due to this, we obtain independent results for nodes. It allows conflicting nodes to be identified and solutions to be identified that will help to improve network and node performance. Another important advantage of the simulator is that it enables heterogeneous networks to be estimated, with different hardware and software for each node.
5.2.1 Wireless Network Model

In a wireless network, the physical channel between two nodes is a shared channel, with noise and interference, where we must take into account that the range of the node is limited. As a consequence, we need to determine the visibility and the probability of a successful reception of a packet sent between nodes. For this, and with the goal of performing accurate simulation, the developer must study the WSN deployment zone and define the probability of packet loss among all nodes. This probability data may be calculated by electromagnetic propagation simulation, for example the computer tool Cindoor. Cindoor is an engineering tool for the effective implementation of wireless systems. This probability must include the probability of loss due to noise.

With these defined probabilities, the simulator can know the effectiveness of the links between nodes. If the developer defines a link as 100, it means that the sending node range is not enough to reach the destination directly. We can observe these defined probabilities in Figure 2 represented as “NODE LINK” (Radio link between node 0 and node 1).

5.2.2 Implementation in the Network Simulator

The network is responsible for transmitting packets to the destination. When a node sends a packet, the network adds the packet to the transmission queue that is sorted by time of arrival at the reception node. When the simulation times match the time of arrival of the packet, the wireless network pops the packet and generates a real random number between 0 and 100. If the probability of success is higher than this random number, the network transmits the packet to the destination, otherwise, it discards it.

Figure 2. Wireless network model.

6. Attack Modeling Technique

The technique used for modelling a jamming attack in the virtual simulator is based on the introduction of a new special node in the simulation. This special node is the attacker. The main function of this attacker is the introduction of noise in the network. This causes an increase in the packet error rate between nodes. The attacker node is responsible for introducing noise into the links defined. In addition to the defined links, we have to introduce defined noise power. This power is based in the percentage of noise affected packets. The definition of a power of 100% means that the link affected packages will all be lost when the attacker node is running. As we specify in the previous section, there are different jamming strategies, BroadBand noise, Partial Band noise, etc. To simulate these different jamming attack paths, we can specify the number percentage of packets affected.

6.1 Implementation in the Virtual Simulator

New attacker nodes are introduced into the network model of the selected virtual simulator. For this, during the network topology definition, we have to specify the attacker’s nodes as explained.

The network is responsible for processing these special nodes. As we can observe in the Figure 3, when the simulation is running and the packets arrive at the network, the network is responsible for sending or rejecting the packets. This decision is based on the error rate that the users specifies in the network model. For each packet, the network obtains a random number (N in Figure 3) and compares this number with the probability of success. If the probability is greater than this random number, the packet is rejected. The rate of each packet is affected by

Figure 3. Simulation of a Wireless Network with attackers.
Deepak Choudhary

Indian Journal of Science and Technology

Vol 9 (48) | December 2016 | www.indjst.org

the specified attackers. They increase this error probability due to the characteristics of the attacker nodes. In this way, we can reproduce the effect caused by a jammer on a wireless network.

7. UML/MARTE Modeling

The system to be modeled consists of a network of nodes connected by wireless links. Each node is modeled as a UML component, specified by the MARTE stereotype <<HwDevice>> (all the stereotypes used in this modeling methodology are included in the MARTE profile standard).

Each of these nodes or HwDevices has an internal structure. The HwDevice structure is composed of a processor, memories, a sensor, an element to transmit information and a bus to interconnect all the HW components. The processor is modeled as an UML component specified by the stereotype <<HwProcessor>>. Additionally, the processor may have associated cache memories. The processor caches are defined by using the attribute caches of the HwProcessor stereotype, and attaching components specified by the stereotype <<HwCache>>. The processor has an associated RTOS. Typical RTOSs in WSN are FreeRTOS19 and Tiny OS. In this work Free RTOS is used as the node RTOS. A RTOS SW component is supported by means of the <<Scheduler>> stereotype. This association is modeled as a UML abstraction specified by the stereotype <<allocate>>.

The memories are modeled by a stereotype, <<HwRAM>>, and the internal communicating element is modeled as a <<HwBus>>. Finally, the internal sensor is modeled as a <<HwSensor>> component and the information transmitter as a <<HwI_O>> component.

The HW/SW architecture of a node is modeled in a UML composite structure diagram associated with the HwDevice node components. The HW/SW architecture is composed of instances of the UML component specified by the previously-mentioned set of MARTE stereo types. Additionally, each node has associated applications to be executed. An application is modeled as a UML component specified by the stereotype <<RtUnit>>.

The allocation of this application is captured by a UML abstraction specified by the stereotype <<allocate>>. Nodes with different HW/SW architectures can be considered in the network specification.

From HwDevice node components modeled as Figure 4 shows, the designer can specify the node network. The nodes are interconnected by wireless links. In order to simulate this communicating mechanism, a model of a wireless connection should be included. For that purpose, a network component that models the communication media (in this case, the air) is included in the model. To model the net, a UML component specified by the stereotype <<HwMedia>> is used. The network has a set of channels in order to enable multiple connections among the network nodes. Specifically, in the wireless communication the number of channels is 14. Figure 5 shows the modeling of the HwMedia channels. A channel is modeled by a <<Communication Media>> component.

Each HwDevice node is connected to the net through an UML component specified by the stereotype <<HwEndPoint>>. These HwEndPoint components model the network interfaces required by the simulation tool in order to enable the connection of components to a network. Figure 6 shows the specification of the node network in order to enable the simulation. The network consists in nine interconnected nodes (only four nodes are shown in the figure). In addition to the node network model structure, composed of the HwDevice node instances, the net and the set of net interfaces, the ports of the network component are specified by the stereotype <<Shaped>>. This stereotype enables the identification of a specific node that is connected to the net. This information is used by the simulation tool in order to establish the routing of the packets to be transmitted. As is
Encryption Techniques for Different Introducer’s Attack in Wireless Sensor Networks

Information flow represents the physical channel in normal conditions and the other information flow represents the physical channel when a jamming attack is taking place. Both information flows are specified by the stereotype <<PaCommStep>> that defines a set of non-functional properties (NFPs) for the communication. The PaCommStep attribute used for this methodology is probability which defines the probability of the data transmitted by the source of an information flow being received by the target of this information flow. The probability attribute is typed as NFP_Real. The notation to define the value of the probability attribute should be (value=“value”, Stat Q=“percent”). In the case of defining the communication probability under a jamming attack an additional PaCommStep attribute should be specified. The attribute cause defines the event that modifies the transmission conditions of the virtual channel mmStep under the jamming attack. Therefore, a PaCo describes the communication between two nodes when no jamming attack affects the nodes’ link. The other PaCommStep information flow denotes that the communication between the nodes is under a jamming attack, which implies a reduction in the probability of the data transmission being successful.

Thus, a node can only establish data communication with a limited set of nodes. This set of nodes is the local environment of a specific node. In principle, the modeling in Figure 6 enables the automatic import of packet loss rate data for each node-to-node link, once the names in the UML network model correspond to node names within the wireless sensor network simulator. Additionally, the UML model enables the capture of specific packet loss rate data for specific nodes. This data overrides data imported from the wireless sensor network simulator, and it can be parameterized. Therefore, the automatic generation of a parameterized model of the WSN can be established, facilitating the impact estimation of packet loss rate at specific links. This local environment is composed of the node instances which a physical channel can be established by a particular node. The physical channels established by the particular node with their local environment are defined by two characteristics, the probability of not establishing the communication and the probability of not establishing the communication when the physical channel suffers a jamming attack.

The physical channels are modeled by two UML information flows. The direction of information flows denotes which node is the source or target of the data to be transmitted. The UML model enables the capture of specific packet loss rate data for specific nodes. This data overrides data imported from the wireless sensor network simulator, and it can be parameterized. Therefore, the automatic generation of a parameterized model of the WSN can be established, facilitating the impact estimation of packet loss rate at specific links. This local environment is composed of the node instances which a physical channel can be established by a particular node. The physical channels established by the particular node with their local environment are defined by two characteristics, the probability of not establishing the communication and the probability of not establishing the communication when the physical channel suffers a jamming attack.

The example in Figure 8 shows “node 8” and the different shown in Figure 6, the node network model is composed of instances of the node components connected to a network. Although not all the nodes can transmit information to any other node in the network that is modeled, always there is a physical inter-connection among all network nodes. In practice, this means that only a set of nodes from the entire network are visible for each node.

Thus, a node can only establish data communication with a limited set of nodes. This set of nodes is the local environment of a specific node. In principle, the modeling in Figure 6 enables the automatic import of packet loss rate data for each node-to-node link, once the names in the UML network model correspond to node names within the wireless sensor network simulator. Additionally, the UML model enables the capture of specific packet loss rate data for specific nodes. This data overrides data imported from the wireless sensor network simulator, and it can be parameterized. Therefore, the automatic generation of a parameterized model of the WSN can be established, facilitating the impact estimation of packet loss rate at specific links. This local environment is composed of the node instances which a physical channel can be established by a particular node. The physical channels established by the particular node with their local environment are defined by two characteristics, the probability of not establishing the communication and the probability of not establishing the communication when the physical channel suffers a jamming attack.

The example in Figure 8 shows “node 8” and the different shown in Figure 6, the node network model is composed of instances of the node components connected to a network. Although not all the nodes can transmit information to any other node in the network that is modeled, always there is a physical inter-connection among all network nodes. In practice, this means that only a set of nodes from the entire network are visible for each node.

Thus, a node can only establish data communication with a limited set of nodes. This set of nodes is the local environment of a specific node. In principle, the modeling in Figure 6 enables the automatic import of packet loss rate data for each node-to-node link, once the names in the UML network model correspond to node names within the wireless sensor network simulator. Additionally, the UML model enables the capture of specific packet loss rate data for specific nodes. This data overrides data imported from the wireless sensor network simulator, and it can be parameterized. Therefore, the automatic generation of a parameterized model of the WSN can be established, facilitating the impact estimation of packet loss rate at specific links. This local environment is composed of the node instances which a physical channel can be established by a particular node. The physical channels established by the particular node with their local environment are defined by two characteristics, the probability of not establishing the communication and the probability of not establishing the communication when the physical channel suffers a jamming attack.

The physical channels are modeled by two UML information flows. The direction of information flows denotes which node is the source or target of the data to be transmitted. One information flow represents the physical channel in normal conditions and the other information flow represents the physical channel when a jamming attack is taking place. Both information flows are specified by the stereotype <<PaCommStep>> that defines a set of non-functional properties (NFPs) for the communication. The PaCommStep attribute used for this methodology is probability which defines the probability of the data transmitted by the source of an information flow being received by the target of this information flow. The probability attribute is typed as NFP_Real. The notation to define the value of the probability attribute should be (value=“value”, Stat Q=“percent”). In the case of defining the communication probability under a jamming attack an additional PaCommStep attribute should be specified. The attribute cause defines the event that modifies the transmission conditions of the virtual channel mmStep under the jamming attack. Therefore, a PaCo describes the communication between two nodes when no jamming attack affects the nodes’ link. The other PaCommStep information flow denotes that the communication between the nodes is under a jamming attack, which implies a reduction in the probability of the data transmission being successful.

The attribute cause of the <<PaCommStep>> is typed as a Ga Workload Event element. Thus, a UML component specified by the stereotype <<GaWorkloadEvent>> should be included in the model. This GaWorkload Event represents a jamming event. This GaWorkload Event can be specified by defining the period between two jamming events; specifically in the attribute pattern (i.e. periodic(20 ms)).

The GaWorkload Event can represent different jamming attacks. Figure 7 shows the modeling of the different jamming attacks described in Section 4. The different jamming attacks are specified by an UML constraint associated with the corresponding GaWorkload Event; only in the case of BroadBand Noise, no UML constraint should be defined (Figure 7a). The UML constraint defines which network channels are under the jamming attack. In Figure 7b, the UML constraint defines that the jamming attack is applied to a continuous set of channels (jamming->(2…12)), specifying Continuous and Partial Band Noise jamming. Figure 7c shows a discrete set of channels where the jamming attack is applied, which defines the Discontinuous Partial Band Noise. Finally, Figure 7d specifies a NarrowBand noise jamming attack. The example in Figure 8 shows “node 8” and the different
as a \textit{GaWorkloadEvent}; in the case of “node 1” and “node 3” “jamming\textit{NarrowBand}” and “jamming \textit{Discontinuous PartialBand}”, respectively. The physical channel established between “node 8” and “node 6” is a different case. In this case, the channel link suffers two different jamming attacks, the “jamming \textit{BroadBandNoise}” and “jamming \textit{Continuous Partial Band}”.

From the probability values of a physical channel, the probability of not establishing the communication can be extracted to enable the characterization of all network node communication which enables the network simulation.

8. Bridging UML/MARTE Model-simulator

In order to provide the information captured in UML/MARTE model to the simulator a set of XML files have been defined. The XML file “network\_platform” defines the HW components used for modeling the node structures, the internal architecture of each node network (modeled as in Figure 4) and declares the network component, defining the number of nodes connected and the number of network channels.

A second XML file, named “network\_archi”, describes the network architecture, identifying the nodes with an ID (modeled by the \textit{Shape} stereotype) and the different connections of each node with the other nodes of the network. These connections are characterized by the probability of a successful connection when no jamming attack. It is modeled as Figure 8 shows. Finally, the XML file “jamming\_attack” defines the different jamming attacks captured in the model (Figure 7). Each jamming attack is characterized by the period between attacks and the different node links attacked. These node links are defined by the node source, the node target and the probability of establishing the communication. Finally, the XML file “jamming\_attack” defines the different jamming attacks captured in the model (Figure 7). Each jamming attack is characterized by the period between attacks and the different node links attacked. These node links are defined by the node source, the node target and the probability of establishing the communication. Finally, the XML file “network\_platform” file.

![Figure 7. Specification of different jamming attacks.](image)

![Figure 8. Model of the local environment of node 8 and the physical channels.](image)
The virtual platform reads this XML file and generates the HW components with their HW architecture and the SW stack for each network node. Then, the nodes are interconnected, extracting the information from the “network_archi” file. The simulator obtains each link rate and builds a square matrix with these probabilities, facilitating the rate consultation for each link when the simulator is running. The square matrix enables a faster computational access than through the XML file directly. The network can access the link rate between the node “i” and “j”, extracting the position of “i” and “j” from the matrix. The third input to the virtual simulator is the “jamming_attack” file. This is an optional file, because it can simulate the network performance without attackers.

This is the reason why the arrow of this XML file in Figure 9 is discontinuous. If the file is included in the execution of the virtual simulator, it extracts the jammers defined with their information and generates a new special node (explained in the section 6 of this paper) with the information about the model.

Once all the information is extracted by the virtual simulator from these XML files, the simulation can start and the performance reports can be calculated.

8.1 Automatic Generation

In order to implement the MARTE-based methodology and to automatically obtain the XML files, a first code generator prototype has been implemented. The code generator has been written as a set of generation templates written in the standard MTL language. This makes the generator open and portable for different generation engines. The development has been done through Acceleo, a code generation framework, fully integrated in Eclipse.

9. Experimental Results

In this section, we present the results obtained from the virtual simulator using our attack simulation technique. For this, we have designed a small network with nine nodes interconnected as can be seen in Figure 10 (a fragment of the UML/MARTE model can be seen in Figure 5). The probability of losing a packet between two interconnected nodes is 5% in all cases. We do not use a real case example because the purpose of this paper is not to demonstrate the virtual network simulator but the UML/MARTE-based simulation methodology. The nodes will have the same Hardware Architecture and software running. The platform model of the nodes is composed of an ARM 926, a memory, a thermal sensor and an 802.15.4 Zig Bee transceiver. All these components are interconnected by a system bus. The Zig Bee module is configured during the simulations with default parameters. We can find these parameters in. The most important parameters for the simulation are the number of Mac Retries (10), the Baud Rate (9600bps) and the Power Level (18dBm).

The application model consists in software that reads the sensor every 5 seconds and sends this information to its neighbor nodes. This transmission is not in broadcast mode, but in point-to-point transmission.

9.1 Simulated Results

We simulate four attacks on “node 8”. As Figure 6 and Figure 10 show, “node 8” is connected to “node 1”, “node 3” and “node 6”. Each connection between “node 8”...
and the other nodes in its environment are under different jamming attacks. This jamming configuration affects the time and therefore, the number of affected packets to be transmitted. In Figure 10, the attackers are represented as “J” with a number. Jammer 1 consists in a NarrowBand Jammer attack. Jammer 2 introduces Discontinuous Partial Band Noise. Lastly, Jammers 3 and 4 carry out BroadBand and Continuous Partial Band attacks, respectively. The characteristics of these attackers are defined in Figure 8. Figure 11 shows the increment between the attacked nodes and the same nodes without jamming attack. As can be seen in this graph, the energy consumption increment in node 1 and node 3 is similar. The reason for this is that the attackers are configured with the same power and the probability of success is similar. The increment in node 6 is problematic.

The reason for this is that the percentage of the packets affected by attackers is higher, because the configurations of the attackers are more dangerous for this link. The combination of jammer 3 and 4 on the same node produces more damage.

10. Conclusions

In this work, we present an innovative methodology for modeling and simulating secure Wireless Sensor Networks. We use UML/MARTE models to specify the WSN to be simulated. From of these UML/MARTE models, automatic generation can be implemented, generating a set of XML files. These XML files are the inputs for the simulator. The simulator enables one of the biggest problems in this type of networks to be simulated, namely, attacks; specifically, jamming attack, which is one of the most problematic attacks. With these estimations, the developer can take accurate decisions in order to optimize the application software and the network deployment in early design stages.

The next step for this research is the modeling of new types of attacks in the virtual simulator to enable the simulation of all kinds of attacks.

11. References