Wireless Sensor Network Simulation for Security and Performance Analysis

A. Díaz, P. Sanchez
University of Cantabria
{adiaz, sanchez} @teisa.unican.es

J. Sancho, J. Rico
TST Sistemas
{jsancho, jrico} @tst-sistemas.es

***Abstract*— During the last years, Wireless Sensor Networks (WSN) have been deployed at an accelerated rate. The complexity and low-power requirements of these networks have also been growing. Therefore, WSN developers are beginning to require efficient methodologies for network simulation and embedded SW performance analysis. These tools should also include security analysis. This security analysis has to evaluate the vulnerability of a WSN to the wide variety of attacks that these networks could suffer. WSN attacks could also affect power consumption and performance of the node's software, thus security analysis has to be integrated into a complete performance analysis framework. This work proposes a methodology to simulate the most common and dangerous attacks that a WSN can suffer nowadays. The impact of these attacks on power consumption and software execution time are also analyzed. This provides developers with important information about the effects that one or multiple attacks could have on the WSN, helping them to develop more secure software.**

***Index Terms*— WSN, Attack Simulation, Power Consumption, Performance Analysis, Security.**

I. INTRODUCTION

In recent years, smart environments are increasingly being deployed in building, military, health, ecological, industrial, and transportation applications. These environments are based on data acquisition from the real world, communication of these data to processing centers and generation of information-based services. The information used by smart environments is provided by WSNs, which are responsible for monitoring and recording physical or environmental conditions and communicating the collected data to a central location. These WSNs are a group of spatially dispersed-self-powered sensing devices (node). Commonly, this kind of network consists of a large number (from tens to 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 [1].

Every WSN has specific nodes that must comply with system, network and sensor requirements. During recent years, a common requirement of WSN specifications is security assessment. Due to the wireless and unattended nature of these devices, an improvement in the embedded security mechanisms is currently necessary. Although some of the known security threats to traditional networks are equally applicable to a WSN, the latter has specific vulnerabilities due to its shared, unsafe and unprotected communication channel, deployment in hostile and unattended environments, limited computational resources and bandwidth, strict power management policies and system complexity. Additionally, an attacker has direct access to the network nodes and there are a huge number of possible ways to attack the node/network.

The required security level may vary from one application to another according to the importance of the information that is being obtained and/or exchanged. For this reason, at the design phase it is important to identify the security weaknesses of the WSN. Furthermore, understanding the potential effects of some typical attacks on a node or the entire network helps to prevent other problematic vulnerabilities. This is of great value for designing the node's embedded software and/or the complete WSN system.

The design of security services in WSNs must take into account their memory, computational capability and resource availability. Additionally, energy resources are also limited. Normally, power consumption is the greatest constraint in WSNs. Sensor nodes are commonly battery-powered devices, in which their useful-life depends on their battery life. These nodes are often placed in hostile environments with difficult node access after deployment that affects in-field battery replacement. Additionally, the high number of deployed nodes and their distribution within the network also affect battery replacement, thus the power consumption impact of the security services must also be taken into account. Moreover, one of the main effects that WSN attacks commonly produce is the increase in power consumption in the attacked nodes, with a reduction of their useful-life and a loss of communication performance.

The paper is organized as follows. Section 2 presents an analysis of the state of the art. Section 3 describes the proposed WSN simulation techniques. Section 4 deals with the models of attack. Section 5 reports experimental results with several WSNs and Section 6 states the conclusions.

II. STATE- OF-THE-ART

An attack can be defined as an attempt to gain unauthorized access to a service, resource or information. It could also be an attempt to compromise integrity, availability, or confidentiality of a system. In this work, an evaluation of the most typical attacks that a WSN can suffer has been performed. The analysis is based on vulnerabilities described in [2]. The typical WSN attacks can be classified into 11 categories: jamming attack, collision attack, overwhelm attack, black hole attack, interrogation attack, hello flood attack, misdirection attack, sybil attack, application attack, tampering attack, DoS attack.

Current WSN simulation tools do not offer the possibility to simulate typical attacks that these networks can suffer.

978-3-9815370-0-0/DATE13/©2013 EDAA

This works has been funded by Spanish MICINN under EUI2010-04255 and TEC2011-28666-C04-02 projects.

Reference [3] presents an overview of current WSN simulation frameworks. NS-2[4] and OMNET++[5] are discrete event network simulators that have been used to model WSNs. Another framework, TOSSIM[6], is a bit-level discrete event simulator and emulator of TinyOS[7].

As mentioned above, one of the main objectives of the attacks is to increase the power consumption of the node. To make an accurate estimation of the attack's effects, it is important to use a performance analysis framework that provides power consumption and execution time estimations. Some of the commented simulators can provide WSN simulation, RTOS support and power/execution time estimation but, as far as we know, there is no simulation framework that integrates these features and attack simulation.

This paper proposes a virtual platform that integrates all these features (WSN simulation, RTOS integration and performance analysis) and attack simulation. It is able to execute the same code that will be executed in the physical node and to generate the same network traffic that a real node, although this paper is focused on attack simulation. The WSN virtual platform supports the open-source FreeRTOS operating system and includes power and execution time estimation.

III. SIMULATION TECHNIQUE

A. HW/SW Co-Simulation

The WSN virtual platform presented in this paper is based on the native-simulation approach detailed in [8]. A virtual platform is a software model of the WSN that enables system simulation. It includes models of the main elements of a WSN: network model and node model. The node model integrates processors, memories, RF-transceivers and sensors.

In a native-simulation-based virtual platform, the application source code is annotated with performance-oriented code. This new code models HW platforms and network characteristics. The HW models include information about the execution time and power consumption of the processor instructions, functional and power behavior of RF-transceivers, memory and bus access delays. Cache and RTOS models are also included in the HW and firmware models.

The annotated code is compiled by a native compiler of the host computer in which the code will be executed (simulated).

B. Virtual WSN Platform

This paper extends the virtual platform in [9] to attack simulation. The original virtual platform enables the estimation of the power consumption of each WSN node, which is a very relevant feature to calculate the damage of an attack. The network model of the virtual platform is based on packet-loss probabilities: for every pair of nodes, a packet-loss probability has to be defined. It identifies the probability that a packet sent from one node is not received by another. For example, a 100 packet-loss probability means that all the transmitted packets are not received by the second node. This probability data may be calculated by electromagnetic propagation simulation.

Once the packet-loss probabilities define the network topology and wireless-channel quality, the network model is responsible for managing the node's transmitted and received packets.The network model is quite simple: when a node sends a packet, the network model adds the packet to the transmission queue that is sorted by time of arrival at a reception node. All possible in-range reception nodes (packet-loss probability < 100) are considered. When the simulation times match the time of arrival of the packet, the wireless network pops the packet from the transmission queue and generates a real random number between 0 and 100 (N in Figure 1). If the packet-loss probability between the transmitter and receiver nodes is higher than this random number, the network model will transfer the packet to the destination node. Otherwise, the packet will be discarded.

IV. ATTACK MODELING TECHNIQUE

In order to simulate attacks on the virtual platform, this paper proposes to introduce special new nodes (attackers) in the WSN model. These new nodes model the attackers with basic simulation models for speed-up the simulation. Three basic types of attacker nodes are defined; "Link-Noise node", "Fake packet injection node" and "Direct Attack node".

A. Link-Noise node

A Link-Noise node dynamically modifies the packet-loss probability between nodes (reduction of the communication link quality), thus packets could be received with more difficulty or even fail to reach their destination.

LinkNoise (links[], power[],numPackets, time[], typePackets) (1)

As was shown in Equation 1, the link-noise node enables the definition of several parameters:

- **Links:** List of communication links or node-pairs that are affected by this attacker node.
- **Power**: List of noise weighting that will be applied to every link defined. This specifies the percentage of packet-loss probability that will be added to the original probability.
- **NumPackets:** Percentage of packets affected by the increased packet-loss probability.
- **Time:** It defines the ranges of time in which the attacker affects the network. The attacker can be turned on and turned off many times during the simulation
- **TypePackets**: The attack will only be active for specific packet types, enabling the simulation of selective attacks.

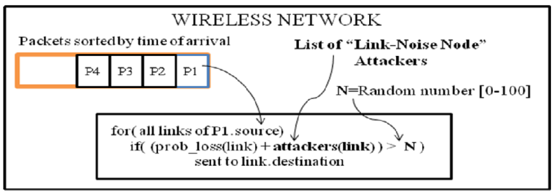

Fig. 1. Simulation with Link-Noise attackers

In order to simulate "Link-Noise" nodes efficiently, the previously commented WSN network model has been modified. Basically this attacker modifies the packet-loss probability for certain packet types during pre-defined periods of time. The network simulation model includes the new probability as Figure 1 shows. When a packet has to be transferred to the receiver node, the reception probability will include the original link probability and the additional noise produced by attacker (prob_loss and attackers in Figure 1).

B. *Fake packet injection node*

This attacker node introduces fake packets into the network with different motivations. The packets are received by the nodes because their structure is formally correct. Equation 2 shows the definition of this type of attackers.

FakeInjec(freq, typePacket, time[], nodesDestine[], broadcast) (2)

The packet pay-load depends of the attacker parameters that model the attack mode, injection frequency, fake packet types, etc. These parameters are:

- **Frequency:** It defines the fake packet rate or number of packets per second that are injected into the network.
- **TypePackets**: It defines the type of packets that the attacker injects. As for example: ACK, RTS and data packets.
- **Time**: It defines the range of time in which the attacker is running.
- **NodesDestine:** It specifies the receptors of the packets.
- **Broadcast**: Each packet will be sent to all nodes.

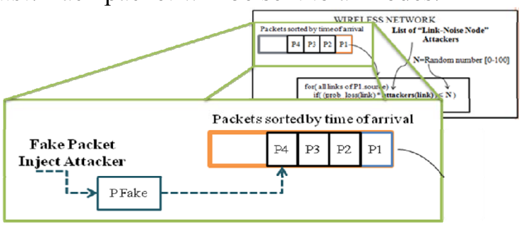

Fig. 2. Simulation with Fake Packet Injection attackers

In order to simulate this attack, the attacker node injects new packets into the transmission queue of the network model. Once the packet is inserted in this queue, the network transmits these fake packets as if they were genuine. Figure 2 shows this attack in the network model.

C. *Direct attack node*

The direct attack node modifies the node's embedded software. It models the programming of a genuine node by a fake program. The attack definition includes the new application to be downloaded to the node and its network parameters (packet-loss probabilities).

D. *Attacks model and the vulnerabilities identified*

The relationship between the proposed attacks and the attacker model in Section 4 will be explained in this sub-section. Table 1 shows the relationship among the identified WSN vulnerabilities and the proposed attacker model. These models cover all the identified vulnerabilities. For example, the "Direct attack node" enables the modeling of the swamping and application attacks in Section 2.

As can be seen in Table 1, a Link-Noise node can model different attacks: Jamming, Collision, Black Hole and Path-based attacks. Different parameter values of the attacker node configuration enable different types of attacks to be modeled. For example jamming and collision attacks can be modeled with the same attacker node (Link-node) but with different parameters. These parameters are not only used to define attacks, but also to define attack strategies. For example, in the Jamming attack, there are multiple strategies [10]. If there is a Narrow-Band Noise jamming attack [10], the attacker could be modeled as a link-noise node with 10% affected packets.

	Types of attacks		
Link-Noise node	Jamming	Interrogation	Fake packet injecto
	Collision	Hello Flood	
	Black Hole Attack	Misdirection	
	Path-based DoS Attack	Overwhelm Attack	Direct attack
Link + Injector	Sybil Attack	Application Attack	

Table 1 Implementation nodes for each attack

Another vulnerability that can be modeled by this attacker node is Collision attack. This attack may be implemented in two ways: the smart mode (the attacker knows the packet transmission channel) or the brute-force mode (the packet corruption probability is directly proportional to the number or wireless channels). Another example is the interrogation attack. It is simulated with a Fake-Packet injection node. In this case, it is necessary to configure the attacker node to continually send CST and RTS packets. Sybil attack modeling is a special case, because it is necessary to use two attacker nodes. The Noise-Link node isolates the attacked node and the Fake Packet injector node inserts the altered packets.

In Section 2, the tampering attack is mentioned although it is not shown in Table 1.This is because it does not affect the operation of the node, but consists of stealing information.

V. Case Study: Attacks over WSN

In order to evaluate the proposed methodology, two examples of attacked WSNs have been proposed (Figure 3 and 4). Both WSNs are composed of nine nodes. One of the nodes is a Gateway that communicates the WSN with internet using an additional modem (for example GPRS). All the nodes have the same Hardware Architecture and similar embedded software. The platform model of the nodes has an ARM 926 processor, a memory, a temperature sensor and a 802.15.4 transceiver. All these components are interconnected by a system bus. The embedded application software reads the sensor every 30 seconds and sends the information to a gateway node. The gateway node analyzes the information and, if it detects an anomalous measurement, it will activate alarms and report the event to a GPRS network. The network uses a point-to-point protocol.

In the first example (Figure 3) the network has a mesh topology. Each node takes its measurement and sends it to the Gateway node. The percentages on the red lines represent the packet-loss probabilities of the wireless channel. If there is no red line between two nodes, it will be assumed that the packet-loss probability is 100% (no direct connection).

The second network (Figure 4) has a linear topology. Each node reads the sensor and sends the information to the next node. The information message contains the current sensor reading and the previous-node's sensor readings. The red lines

	Mesh Network				Linear Network			
	Gateway	*Node [0-6]*	*Node 7*	*Total*	*Gateway*	*Node [0-6]*	*Node 7*	*Total*
Without attacks	3.614 J	0.713 J	0.713 J	9.319 J	0.563 J	1.598 J	1.598 J	13.354 J
Collision attack	-28.01%	+192.95%	+192.95%	+108.92%	-9.48%	0%	+90.56%	+9.4 %
Interrogation attack	333.51%	0%	0%	+90.65%	+1390.9%	0%	0%	+65.02%
Sybil Attack	0%	0%	+563.2%	+68.21%	0%	0%	314.53%	+20.44%

Table 2 Simulation results

model the possible wireless communication channels and the packet-loss probability.

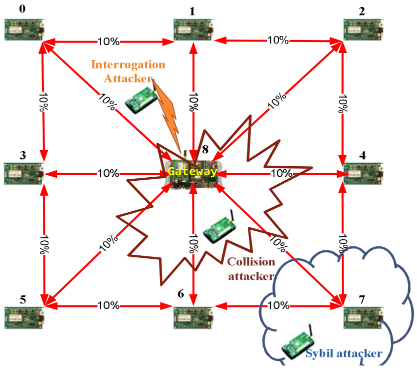

Fig. 3. Wireless mesh Sensor Network

In both cases, 3 attacks have been modeled in 3 different simulations (one attack per simulation). In this study case, it is only considered three relevant attacks due to the large variety presented in the Section 2, but the virtual simulator allows the simulation of the described attacks. The first attack is an intelligent collision attack on the gateways nodes with an effectiveness of the 60%. The second attack is a Sybil attack on node 7 and the third attack is an interrogation attack on the gateway nodes. The objective of these simulations is to estimate which attacks are more problematic (in terms of energy consumption) for each network. These attacks are shown in Figures 3 and 4.

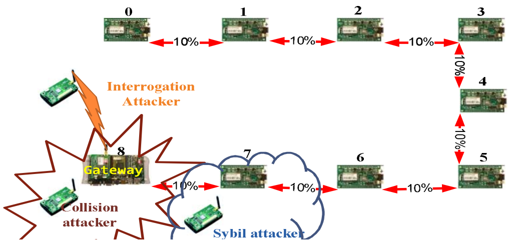

Fig. 4. Wireless linear Sensor Network

Table 2 shows the power consumption of each node in both networks. Row 1 shows the power consumption without attacks and the following rows present the increase of power consumption produced by each attack. As it can be observed in Table 2, the total power consumption of the Linear Sensor Network is 30% higher than the Mesh Sensor Network. This is due to the increased workload of the sensor nodes, because they must communicate with their neighbors. In contrast, the gateway consumption is sometimes lower (eg. -9.48%), because it receives fewer packets. In the collision attack case, the gateway reduces the power consumption because its workload is reduced since the attack decreases the number of packets received. However, when the attack effects on the networks are compared, it can be seen that the mesh network is more affected than the linear network, in terms of consumption. For example, the power consumption of the mesh network is doubled while the linear network consumption is slightly affected by the collision attack. In the Sybil-attack case, the linear network also has a lower power consumption increase. These results do not mean that the linear network is more secure than the mesh network as the reduced power consumption is a consequence of a higher packet loss. In the Sybil attack, the mesh network loses 1/8 packets while the linear one loses all the packet information. Virtual simulator also allows these functional effects of attacks to be detected.

VI. Conclusions

This paper presents a methodology to simulate secure Wireless Sensor Networks. At the early design stages, the WSN developer can take accurate design decisions guided by the estimations that the framework provides.

The proposed technique enables the simulation of attacks in WSNs. The proposed virtual platform includes a HW node, embedded SW, a RTOS, a wireless network and attack models. Three types of attacker nodes have been identified: Link-noise, fake-packet injection and direct attack nodes. These models cover most of the WSN's identified vulnerabilities. These vulnerabilities have been classified into 11 classes in this work.

Experimental results demonstrate that the proposed technique is able to analyze the functional and power-consumption impact of attack on WSNs.

References

[1]Kavitha, Sridharan,"Security vulnerabilities in wireless sensor networks: A survey." Journal of Inf. Assurance and Security. 2010
[2]Mohammadi, Jadidoleslamy, H. "A Comparison of Link Layer Attacks on WSN" Int. journal on applications of graph theory in wireless ad hoc networks and sensor networks.2011
[3]M. Mekni, B. Moulin, "A survey on sensor webs simulation tools", Int. Conf. on Sensor Technologies and Applications, 2008.
[4]NS-2, "The Network Simulator", 2007
[5]OMNeT+-, "www.omnetpp.org" 2012
[6]Levis, Lee "TOSSIM: A Simulator for TinyOS Networks", 2003
[7]http://www.tinyos.net 2012
[8]Posadas, Castillo, Quijano, Fernandez, Villar, Martinez, "SystemC Platform Modeling for Behavioral Simulation and Performance Estimation of Embedded Systems". Modeling for Embedded Systems and Tech.: Applications for Design and Implementation 2010
[9]Diaz, A., Diego, R., Sanchez, P. "Virtual Platform for Wireless Sensor Networks" Euromicro Conf. on Digital System Design 2012
[10]Pelechrinis, Iliofotou, Krishnamurthy "Denial of service attacks in wireless networks: The case of jammers" IEEE Comm. Surveys and Tutorials, 2011