
Olfa Gaddour*, Anis Koubâa§, Rihab Chaari*, Fernando Royo ‡, Nada Al-Elaiwi ¶, Hanan Al-Soli ¶, Stefano Tenniala §, Hichem Boujelben*

* CES Research Unit, National School of Engineers of Sfax (University of Sfax), Tunisia.
§ CISTER Research Unit, Polytechnic Institute of Porto (ISEP/IPP), Portugal.
¶ COINS Research Group, Al-Imam Mohamed bin Saud University (CCIS-IMAMU), Saudi Arabia
‡ Albacete Research Institute of Informatics, University of Castilla-La Mancha, Spain

Emails: olfa.gaddour@enis.rnu.tn, aska@isep.ipp.pt, rihab.chaari@ceslab.org, froyo@dsi.uclm.es, st-nada.alelaiwi@coins.csrlab.org, st-hanan.alsoli@coins.csrlab.org, sota@isep.ipp.pt, hichem.boujelben@ceslab.org

Abstract—Monitoring of Wireless Sensor Networks (WSNs) is a fundamental task to track the network behavior and measure its performance in real-world deployments. Commercially-available products for monitoring and testing IEEE 802.15.4-compliant Low Power Wireless Personal Area Networks (LoWPANs) are mainly too expensive, and typically require special sniffing hardware. In this Demo paper, we present our tool Z-Monitor, a monitoring and a protocol analyzer solution to control and debug IEEE 802.15.4-compliant LoWPANs. Z-Monitor represents a free and extensible solution for monitoring Zigbee, 6LowPAN and RPL protocols, does not require special sniffing hardware, and provides comparable services to proprietary and commercial products.

I. INTRODUCTION

LoWPANs are typically composed of devices that conform to the IEEE 802.15.4-2006 standard. While IEEE standard 802.15.4 specifies the Physical and Medium Access Control (MAC) layers and underlying services for LoWPANs, upper layers like Network and Application layers are defined by other standards like ZigBee [1], 6LoWPAN [2] and RPL [3]. Despite the fact that ZigBee and 6LoWPAN/RPL are arguably the most important WSN technologies today, very little is available on network monitoring and debugging of these networks. In this Demo, we present Z-Monitor [4], a modular application for monitoring and controlling IEEE 802.15.4-compliant LoWPANs. Z-Monitor provides a convenient solution for researchers and students for developing, debugging and deploying wireless sensor network applications based on IEEE 802.15.4 standard protocol and underlying network protocols (i.e. 6LoWPAN, ZigBee, RPL). Z-Monitor is compatible with the open-source official TinyOS implementation of the IEEE 802.15.4 recently released by the TinyOS 15.4 Working Group [5]. It also provides support for both ZigBee and 6LoWPAN [2], the two mostly used protocols deployed over LoWPANs.

II. Z-MONITOR IN BRIEF

Z-Monitor provides an open source, extensible, modular and user-friendly solution for LoWPAN monitoring. Z-Monitor allows for passive monitoring of IEEE 802.15.4-based networks and for analyzing the network behavior through statistical data analysis. Z-Monitor relies on a particular sensor node acting as a passive sniffer that captures network traffic and redirects it to a user-friendly Graphical User Interface (GUI). The fundamental advantage of Z-Monitor as compared to commercially available products such as CC2420 Sniffer [6], Daintree Network Analyzer [7] and Zena [8] is that it is independent of any special hardware and simply relies on a simple mote to capture traffic.

A component-based approach has been used to design Z-Monitor. The block diagram of the main components is shown in Fig. 1.

![Fig. 1: The Block Diagram of Z-Monitor](image_url)

On the hardware side, the sniffer hardware is simply an IEEE 802.15.4-compliant sensor mote, which passively captures the network traffic. Each received packet is redirected to the serial interface through which the sniffer is attached to forward that packet to the software sniffing threads. The sniffer hardware that we have used is a TelosB mote [9] that implements tknsniffer application available under TinyOS. The tknsniffer application switches the radio chip into promiscuous mode and subsequently sniffs all packets that come along. Z-Monitor collects packets arriving from the USB port, stores them in a buffer, performs parsing and
packet decoding and finally displays parsed frames and outputs network statistics as depicted in Fig. 2.

![Z-Monitor Frame Decoding Interface](image)

**Fig. 2: Z-Monitor Frame Decoding Interface**

### III. EXPERIMENTAL STUDY

#### A. Network Test-Bed and Objectives

We present an experimental study that shows how to perform monitoring and performance evaluation of ZigBee, 6LoWPAN and RPL protocols using Z-Monitor. The objectives of the experimental study are manifold:

- To demonstrate the capabilities of Z-Monitor for network monitoring.
- To validate Z-Monitor tool’s support for various IEEE 802.15.4-based networks.
- To show how Z-Monitor is useful in evaluating the performance of IEEE 802.15.4-based WSNs.
- To present the collection of network statistics using Z-Monitor.

The network topology scenario used in the following experiments is composed of 12 TelosB motes in an indoor environment as presented in Fig. 3. In detail, the demonstration consists of the following components:

1) One sniffer mote (running tknsniffer TinyOS application);
2) One Base Station (playing the role of the PAN coordinator, e.g., running IPBaseStation TinyOS application for BLIP [10] or uip6-bridge [11] for Contiki that does the bridging to the nodes running uIPv6);
3) 10 identical nodes distributed around the Base Station;
4) A Notebook with installed TinyOS 2.x operating system.

Currently, Z-Monitor can be used to monitor and analyze Open-ZB [12], BLIP, uIPv6 and ContikiRPL [13] implementations. In fact, this demonstration shows how Z-Monitor performs the parsing and decoding of all IEEE 802.5.4-2006 standard frames, the evaluation and comparison of two well-known 6LoWPAN implementations, i.e., uIPv6 on Contiki and BLIP on TinyOS, and evaluate the performance of the recently drafted RPL routing protocol under Contiki operating system.

For the demo, we have chosen two scenarios showing the capabilities of the Z-Monitor tool. They are described below:

- **Scenario 1.** In this scenario we want to demonstrate the parsing and decoding functions of the tool when a Zigbee Cluster Tree is formed using the beacon enabled version of the IEEE 802.15.4-2006 standard. We use one PAN coordinator and 10 end devices. The end devices associate to the coordinator and send a single packet to the coordinator after receiving every beacon. The sniffer mote attached to the laptop will sniff the frames and send packets received via the serial port to the PC to be analyzed by Z-Monitor. The performance of Z-Monitor are compared against a CC2531 Evaluation Module Kit [14], composed by one CC2531DK Dongle USB radio sniffer and the TI Packet Sniffer tool. The parameters to analyze are the packet reception ratio and the decoding function of the tool.

- **Scenario 2.** In this scenario, we want to demonstrate the ability of Z-Monitor to receive 6LoWPAN packets. To this end, we install the UDPEcho application from the TinyOS implementation in the ten nodes. This application provides a UDP echo service. After installing, we can generate traffic by simply sending ping6 commands to each node from the the node attached to the PC with the base station application is installed. We can monitor the traffic in Z-Monitor through the sniffing mote connected to the laptop, and compare its performance (packet reception ratio and the correct decryption of packets) against the WireShark [15] tool. Fig. 4 shows a screenshot of the planned demo that demonstrates the capture and analysis of BLIP packets with Z-Monitor.

---

1This second option will not be shown during the demo, but it is feasible if the final user wants to try
With this demo, we also show how Z-Monitor can be used to measure the performance of the different protocols. With Z-Monitor, users can measure the network convergence time metric of each router node, which is the duration a node spends to join the 6LoWPAN network for all implementations under study (i.e. Open-ZB, BLIP, uIPv6, and ContikiRPL), and this is done through observing the arrival time of the packets. Fig. 5 shows an example of measuring the convergence time of a 6LoWPAN network with Z-Monitor in which we compare the performance of uIPv6 and BLIP implementations. To do so, we measured the time when a node receives a router advertisement message from the Base Station for both the implementations.

We show also through this demo that Z-Monitor can be used also to measure the throughput and the delay of packet transmission.

IV. CONCLUSIONS AND FUTURE WORKS

The proposed demonstration explains the capabilities of our tool Z-Monitor to monitor, analyze protocols and evaluate the performance of COTS WPANs technologies namely IEEE 802.15.4, ZigBee, 6LoWPANs, and RPL protocols. Z-Monitor is compatible with all open-source implementations of these protocols provided by TinyOS and Contiki operating systems.

We are currently working towards extending Z-Monitor to support more advanced features including (1) support of multi-hop topologies through the use of multiple sniffers so that it will be easier and practical to analyze the behavior of large scale networks, (2) extending parsing component to support new COTS protocols implementations such as TinyRPL, which has recently been released, (3) integrating advanced filtering and statistical analysis features.

V. ACKNOWLEDGMENTS

This work is funded by the Deanship of Research at Al-Imam Mohammed bin Saud University under the COINS Research Group. It is also partially funded by the R-Track project under the grant of the National Plan for Sciences and Technology (NPST), and by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04-03.

We also thank Ahmed Dhikil and Mossab Alsania for programming earlier versions of Z-Monitor.

REFERENCES

[10] BLIP implementation. URL http://docs.tinyos.net/tinywiki/