Demo Abstract: Z-Monitor: A Monitoring Software for IEEE 802.15.4 Wireless Sensor Networks

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

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

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`

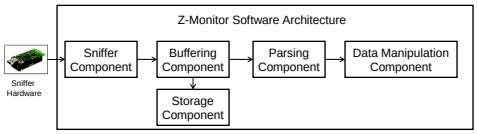

Fig. 1: The Block Diagram of Z-Monitor

application available under TinyOS. The `tknsniffer` application switches the USB port 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 sFig. 2.

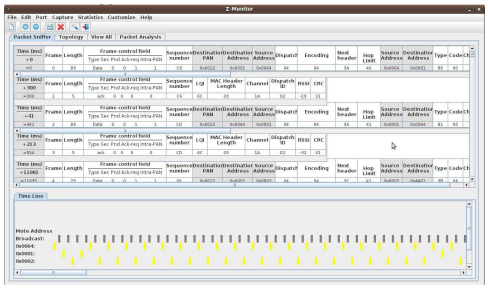

Fig. 2: Z-Monitor Frame Decoding Interface

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 presented in Fig. 3. The demonstration consists of the following components:

1) One sniffer mote (running `tknsniffer` TinyOS application);
2) One Base Station (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 router nodes distributed around the Base Station;
4) Notebook with installed TinyOS 2.x operating system.

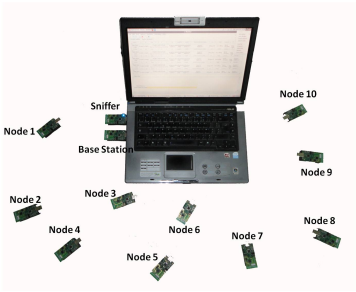

Fig. 3: Experimental Testbed

Currently, Z-Monitor can be used to monitor and analyze ZigBee cluster Tree, BLIP, uIPv6 and ContikiRPL [12] implementations. In fact, this demonstration shows how Z-Monitor perform 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.

B. Demonstration Highlight

The demo scenario is as follows. We deploy 12 TelosB motes in an indoor environment.

TelosB motes are deployed within a single broadcast domain, i.e. a single-hop network. Z-Monitor for multi-hop networks is still underway; therefore we will present results from a single-hop network testbed. The transmission power of nodes was set to - 25 dBm and the frequency channel was set 26. We consider the available open-source implementations of ZigBee and 6LoWPAN protocols namely, the TinyOS IEEE 802.15.4/ZigBee implementation, the TinyOS 6LoWPAN implementation (BLIP), the Contiki 6LoWPAN implementation (uIPv6) without and with RPL support (ContikiRPL).

We install in one mote the sniffer application, in another mote the base station application and in the 10 remaining nodes we install the applications corresponding to the aforementioned implementations. Note that this configuration is suitable for all the applications to be monitored.

During the demo, we have one sniffer mote that must be attached to a laptop running Z-Monitor as it will sniff the received packets by the sniffer and send the packets received via the serial port of the PC to be analyzed by Z-Monitor. Once the nodes are powered, they exchange packets in order to join the network through the Base Station according to the applications. The packets are

captured by the sniffer which sends them to Z-Monitor in order to be analyzed. Fig. 4 shows a printscreen of the planned demo that demonstrates the capture and analyze of BLIP packets with Z-Monitor. It is also possible to run another commercial sniffer in this Demo to compare its output against that of Z-Monitor.

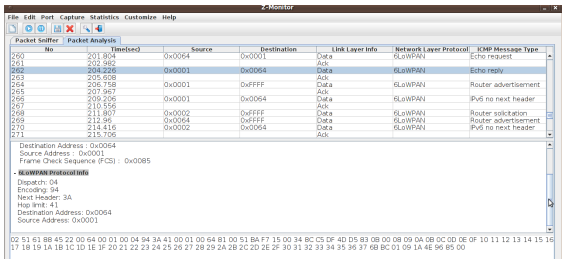

Fig. 4: BLIP Protocol Analysis using Z-Monitor

With this demo, we also show how Z-Monitor can be used to measure the performance of the protocols that it analyzes. 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 both implementations under study (i.e. BLIP, uIPv6 and ContikiRPL, ZigBee Cluster Tree), 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 measure the time when the node receives a router advertisement message from the Base Station for both the implementations.

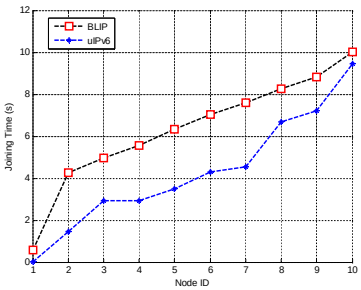

Fig. 5: Convergence Time of 6LoWPAN with BLIP and uIPv6

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

References

[1] Zigbee Alliance, ”ZigBee Specification” (2007).
[2] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler., Transmission of IPv6 Packets Over IEEE 802.15.4 Networks., Internet proposed standard RFC 4944.
[3] T. Winter, P. Thubert, RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, IETF Internet-Draft draft-dt-roll-rpl.txt. 3.
[4] Z-Monitor: A Monitoring Tool for IEEE 802.15.4 WPANs (2011).
URL http://www.z-monitor.org
[5] TinyOS Working Group, url = ”http://www.tinyos.net/”, year = ”2010”.
[6] CC2420 Packet Sniffer, Texas Instruments.
URL http://www.ti.com
[7] Daintree Sensor Network Analyzer.
URL http://www.daintree.net
[8] Zena Network Analyzer.
URL http://www.microchip.com/
[9] TelosB mote platform.
URL http://www.memsic.com/
[10] BLIP implementation.
URL http://docs.tinyos.net/tinywiki/
[11] Contiki Operating System.
URL http://www.sics.se/contiki/
[12] N. Tsiftes, J. Eriksson, N. Finne, O. Fredrik, J. Hglund, A. Dunkels, A Framework for Low-Power IPv6 Routing Simulation, Experimentation, and Evaluation, SIGCOMM10,New Delhi, India (2010) 479–480.