

# ERIKA and Open-ZB: an implementation for real-time wireless networking

Paolo Pagano, Mangesh Chitnis,  
Antonio Romano, Giuseppe Lipari  
Scuola Superiore Sant'Anna  
Pisa (I)

Ricardo Severino, Mário Alves,  
Paulo G. Sousa, Eduardo Tovar  
Polytechnic Institute of Porto  
Porto (P)

## ABSTRACT

IEEE 802.15.4/ZigBee and TinyOS have been playing an important role in leveraging a new generation of large-scale networked embedded systems. However, based on previous experience on the implementation and use of the IEEE 802.15.4/ZigBee protocols over TinyOS, several problems (producing loss of synchronization and even network failures) emerge due to some limitations of this OS, namely related to the lack of task pre-emption and prioritization. When real-time guarantees are required, different software solutions must be used to support real-time services in such networked applications. This being our objective, we implemented the IEEE 802.15.4 protocol over ERIKA, a real-time operating system for resource-constrained embedded systems. The results attained so far, and summarized in this short paper, demonstrate that ERIKA enables reliable network behaviour and improved network performance. This paper outlines the most important aspects of the software implementation and reports comparative experimental results based on real platforms<sup>1</sup>.

## Categories and Subject Descriptors

C.2.1 [Computer Communication Networks]: Wireless Communication; C.3 [Special-Purpose and Application-Based Systems]: Embedded Systems

## 1. INTRODUCTION

The IEEE 802.15.4/ZigBee [1, 2] protocols are being explored to support real-time and energy efficient Wireless Sensor Network (WSN) applications, since they enable timeliness guarantees for data transmission and reception when operating in beacon-enabled mode. To enable this kind of applications, Open-ZB [3], an open source implementation of the IEEE 802.15.4 / ZigBee protocols, was developed over the TinyOS [4] Operating System (OS). TinyOS is one of the most used OSs for embedded resource-constrained wireless sensor platforms. However, TinyOS does not support cru-

<sup>1</sup>This work has been partially funded by FCT under the CISTER Research Unit (FCT UI 608), CMU-PT projects, the EU FRESOR project (FP6/2005/IST/5-034026), and ARTIST2/ARTISTDesign NoEs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.  
Copyright 2009 ACM 978-1-60558-166-8/09/03 ...\$5.00.

cial mechanisms for multitasking such as tasks pre-emption and prioritization, thus it is not capable of guaranteeing a predictable timing behavior of the local computations at node level (and a fortiori at network level).

Therefore, non-preemption and the lack of task prioritization are the major drawbacks of the Open-ZB implementation. In order to overcome this limitation, in this short paper we outline an alternative implementation of the Open-ZB protocol stack over the ERIKA real-time OS [5], that is further extended in [6].

## 2. SOFTWARE IMPLEMENTATION

The implementation of the IEEE 802.15.4 protocols over ERIKA is organized in a layered architecture. Fig. 1 illustrates the overall software architecture. This stack has been implemented for the dsPic platform over the FLEX [7] boards equipped with CC2420 radio transceiver.

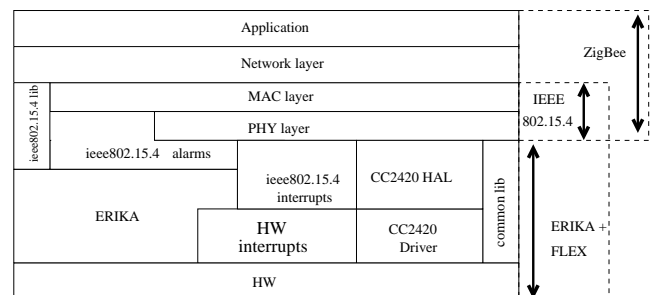


Figure 1: Layered Architecture

All hardware interrupts are handled by the ERIKA Interrupt Service Routines (ISRs). The ieee802\_15\_4 layer contains the code to initialize the hardware timers, as well as the communication between CC2420 transceiver and MCU, and to handle timer and transceiver interrupts. The ERIKA layer is responsible for managing the system hardware resources providing the typical OS services such as Task management, resource access control, interrupt and timer management. Software timer abstractions are provided by means of software counters and alarms. The common lib is a generic library providing some software utilities such as: basic data structures; memory buffer management; debugging helper. The ieee802\_15\_4 Lib implements the PHY and MAC layers of IEEE 802.15.4 standard.

## 3. PERFORMANCE RESULTS

In this section, we compare the performance (throughput and packet delivery ratio) of our platform (ERIKA + Open-ZB on FLEX)

with respect to other popular ones, namely: TinyOS 2.0 + BMAC on Telos-B ; TinyOS 2.0 + Open-ZB on Telos-B.

Our testbed consists of a multi-task firmware with the Constant Bandwidth Rate (CBR) traffic generator scheduled together with the other tasks implementing the network stack services. The payload (104 bytes), and the number of transmitted packets (1000) are pre-programmed as well as the inter-frame period (i.e. the inverse of the traffic rate). We carried out the same experiments on the FLEX both at full speed and at the pre-scaled frequency of 8 MHz to enable a fair comparison between both platforms (FLEX and TelosB). As shown in Figs. 2 and 3, the effect introduced by the CPU speed is not negligible at low rate, up to about 200 Hz.

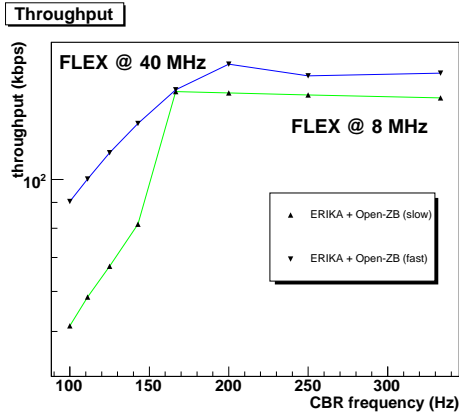


Figure 2: Throughput measurements using ERIKA.

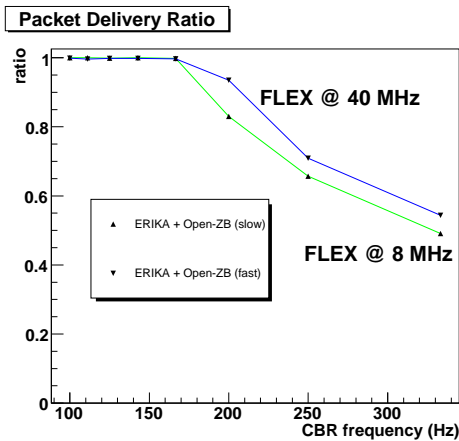


Figure 3: Packet delivery ratio measurements using ERIKA.

We carried out several experiments on the two equivalent platforms, each time increasing the rate of packets sent, and measuring the effective throughput and packet delivery ratio. For better comparison, we ran the same experiments on the FLEX both at full speed and at the pre-scaled frequency. The results are shown in Figs. 4 and 5. The ERIKA solution outcores the TinyOS-based ones at every CBR rate apart from the first point (100 Hz). The saturation of the curve observed around 150 kbps is interpreted as the transceiver maximum sustainable rate since it is very much correlated with the drop in the delivery ratio (i.e. probability of successful transmission).

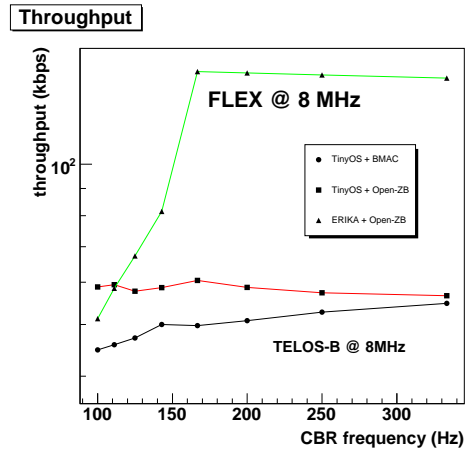


Figure 4: Throughput measurements.

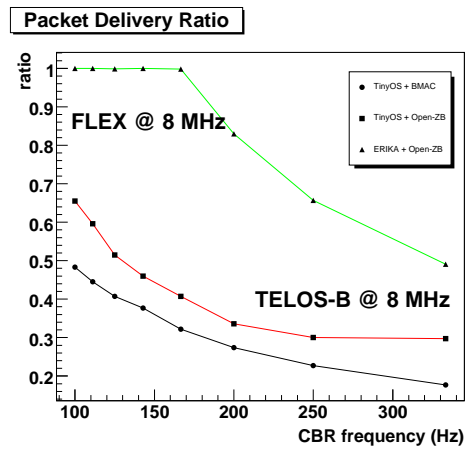


Figure 5: Packet delivery ratio measurements.

## 4. CONCLUSIONS

Our hardware/software platform can achieve a higher throughput and packet delivery ratio with respect to existing solutions based on TinyOS. Given these concrete and promising results, we firmly believe that it is indeed possible to provide support for real-time execution and network transmission in cheap hardware platforms. It is also envisaged to implement the core ZigBee Network Layer functionalities to support multi-hop communications.

## 5. REFERENCES

- [1] IEEE 802.15 WPAN™ Task Group 4 (TG4).
- [2] ZigBee Alliance (2006), ZigBee Specification 2006.
- [3] Open-ZB open-source toolset for the IEEE 802.15.4/ZigBee protocols web site. <http://www.open-zb.net>.
- [4] University of California, Berkeley (CA, USA). The TinyOS Operating System. <http://www.tinyos.net/>.
- [5] The E.R.I.K.A. Enterprise realtime Operating System. <http://www.evidence.eu.com/>.
- [6] P. Pagano, R. Severino, et al. ERIKA and Open-ZB: a tool suite for real-time wireless networked applications. Technical report, HURRAY-TR-031008, October 2008.
- [7] The FLEX evaluation board. <http://www.evidence.eu.com>.