©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Analysis of IEEE 802.15.4 Throughput in Beaconless Mode on micaZ under TinyOS 2

Nelson I. Dopico, Carlos Gil-Soriano, Iñigo Arrazola and Santiago Zazo
ETSI de Telecomunicacion
Universidad Politecnica de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
Email: {nelson,gilsoriano,i.arrazola,santiago}@gaps.ssr.upm.es

Abstract—Study and evaluation of wireless sensor networks (WSN) require modeling of different elements and settling performance boundaries. Available IEEE 802.15.4 models make their own assumptions of the standard within a given scenario. Such assumptions (and interpretations) do not really match the performance of real equipment (nodes) sometimes. Here we propose a model based on one of the most utilized WSN platforms: micaZ. Its operation and timings are described as well so that they can be used for posterior modeling if other scenarios need to be studied. The throughput upper bound of peer-to-peer micaZ-TinyOS links is provided based both on our model and real measurements for the full payload range. The proposed model is compared to both real measurements and a model based on the standard. Results show our model to be accurate and better matching reality for any payload.

1. INTRODUCTION

Certain wireless sensor network applications depend on the channel throughput when it comes about quality of service (QoS) requirements. Thus, some scenarios like [1] or disaster-recovery networks (in which time is a key factor) [2] may benefit from having a model capable to provide them with -at least- boundaries on their performance or, even better, some QoS figures. A number of models have been proposed to measure throughput, loss probability or related issues in IEEE 802.15.4 [8]. Many research works utilize micaZ as a testbed architecture [9]–[13] to the extent that it seems to be one of the most spread ones for research in wireless sensor networks. Therefore, we propose an empirical-based micaZ model that may help researchers and practitioners in their work since, as shown later, available theoretical models may not match real equipment performance. This paper proposes such a model based on a detailed analysis of its timings, components and operation. Unlike other papers which may compare with simulations, we have developed an experiment over the full range of payloads which helps compare model and real world (indeed performed very close to the proposed model). In addition to the model itself, enough information from the micaZ analysis (not usually reported anywhere) is provided so that different models can be developed as well depending on users’ needs. Results provide as well the throughput upper bound achievable on point to point links which is applicable to multipoint-to-point also.

The rest of this paper is structured in six sections. Section 2 provides an overview of existing models and their scope and compares them with the current paper. Section 3 comprises a thorough analysis of the micaZ medium access control (MAC) under TinyOS 2. Section 4 depicts the modeled scenario. Section 5 analyzes theoretical IEEE 802.15.4 and (theoretical and empirical) micaZ links providing a model of the micaZ throughput along with the packet loss probability. Section 6 compares the experiment carried out with both a theoretical IEEE 802.15.4 model and the proposed micaZ model. Section 7 concludes the article.

II. RELATED WORK

Different approaches have been carried out in order to model generic IEEE 802.15.4 links or based-on-mote architecture links. Analysis is mostly based on analytical models or simulations [3]–[7] though some on real test beds [9], [14], [15]. Ref. [16] provides a number of measurements and micaZ real tests of throughputs reached at various payloads. Unfortunately, measured times are missing additional information and the throughput measure technique seems not to be effective as it is measured at a backlogged sender rather than the actual packets received at the sink node.

A number of models have been proposed for slotted carrier sense multiple access (CSMA) on 802.15.4 [4]–[7], but few regarding unslotted (non beacon-enabled) mode [3], [17]. Ref. [3] presents a complete analysis of a scenario composed by a sink node which requests information from a number of sensing nodes; such request triggers an attempt to send data to the sink on every sensing node. The current paper scenario differs from it as the sink node does not trigger any channel access.

A model presented in [17] analyzes a similar scenario as done in the current, whereas the present paper models non-acknowledged traffic and goes a step further detailing such for a real platform widely used by researchers and practitioners: micaZ. In addition to the aforementioned, micaZ operation and timings are described to the extent that they should be sufficient in the event that any other scenario were modeled. Should other architectures be modeled, this may be a useful road map to look at on what the delays and operation of real architectures are.

III. MICAZ ARCHITECTURE

MicaZ is a reference mote architecture [9]–[13] composed of a microcontroller unit (MCU - Atmel ATmega128 microcontroller [18]), a transceiver (Texas Instruments CC2420 [19]) a monopole antenna and a number of peripherals such as leds, flash memory and a 51-pin connector. Microcontroller-transceiver communication is implemented over a Serial Peripheral Interface (SPI) bus and control pins such as Clear Channel Assessment (CCA) and Start of Frame Delimiter (SFD). The bus is used to send and receive packets, command strobes and their replies. Both the bus and its usage can turn out to be a bottleneck under certain circumstances as is depicted later. The MAC implementation [20] on study is provided by TinyOS...
2.1 [21] and it is often referred to as non-slotted IEEE 802.15.4 or beaconless.

A. Operation

A number of operations are carried out before the actual transmission over the air or the data delivery to the microcontroller. Transceiver reception states are described in [19] and are out of the scope of this paper as the TinyOS implementation for reception does not affect the link performance, i.e. it does not behave as a bottleneck. However, transmission operation is detailed as it limits the performance of any IEEE 802.15.4 link based on micaZ.

Any packet transmission is triggered by an MCU decision as a consequence of any event such as a sensor measurement over a threshold, routing of an incoming packet, polling, etc. As soon as the decision is made, a process of transmission over the SPI and channel access starts. First, the SPI is selected, the TXFIFO address is sent and the MAC protocol data unit (MPDU) along with its length are sent over the bus. Then, a random backoff is computed. Ref. [8] sets an initial backoff that ranges from 0 to $2^{\text{BE}-1}$ backoff periods. One backoff period ($\sigma$) lasts for 20 symbols (320 $\mu$s); BE ranges from 0 to 8 and is 3 by default. In the event that the channel is assessed to be busy, subsequent backoffs will increase BE by one up to $\text{macMaxBE}$ which ranges from 3 to 8. In micaZ the initial backoff ranges from 352 to 10240 $\mu$s therefore, BE approximates initially to 5. Congestion backoffs are computed after the first assessment of busy channel as it was done for the regular backoff, however micaZ operates differently congestion backoffs by setting their range from 352 to 2592 $\mu$s.

After the backoff, the MCU reads the CCA pin to assess the channel state. CCA on CC2420 is compliant with the standard [8] as it evaluates the channel over the last 8-symbol periods. If the channel is assessed to be idle, a software delay is performed by calling a timer before issuing a STXONCCA strobe over the SPI bus. Such delay helps the current implementation to comply with the standard as it forces a CCA which comprises an assessment period after the backoff rather than overlapping with it as it would happen if STXCCAON were issued right after the backoff. CC2420 evaluates the channel after receiving the strobe and starts the transmission operation if it is found to be idle. Transmission operation comprises a turnaround time and the physical-layer (PHY) packet as defined in [8].

The entire job is not done once the packet is broadcasted, but the transceiver still has to switch to a state ready to transmit or receive according to the user’s settings. Such delay must be taken into account as it is detailed later.

B. Randomness and backoffs

Randomness on micaZ has been analyzed in order to achieve a more accurate model. MicaZ pseudo random generator is a fast implementation of the Park-Miller Minimal Standard Generator [22] for pseudo-random numbers which uses a 32-bit multiplicative linear congruential generator and a 64-bit temporarily seed. Node ID number is used as a regular seed in TinyOS 2; consequently entropy tests have been performed over a subset (1-255) of the ID numbers. Min-entropy as in [23] has been used as an estimator. Min-entropy is an estimator more restrictive than Shannon entropy (actually the bound which ranges from 0 (null variation) to 1 (uniformly distributed) and it is calculated over the worse case - the most likely observation. Tests were run over $2^{25}$ samples on every node ID from the random 16-bit function offered by TinyOS 2. For every numerical series both entropy and average were computed, hence we obtained 255 values for each estimator. Entropies ranged from 0.6652 to 0.6875 (Fig.1a) and average values fell between 32716 and 32815 - Fig.1b.

The expected average value ($E[\tilde{T}_{\text{backoff}}]$) for the micaZ initial backoff is 5290.28 $\mu$s supposed random values are uniformly distributed and allowing for the fact that some will be more likely as a modulus operation is performed on the output of the 16-bit random function. From the aforementioned tests, the average backoff value obtained is 5289.88 $\mu$s which is very close to the theoretical one. Information on backoff min-entropy and backoff average-value distributions can be found in Fig.1c and Fig.1d.

C. Timing

Delays and timings are hereafter described. Capital letters refer to variable times or lengths while lower cases apply to fixed values. For a better comprehension, see Fig.2. The upper signal on Fig.2 is the current consumption profile which indicates the ongoing state. For our test, just the transceiver has an impact. Channel 1 (the signal just below the numbers) switches upon every state transition (as referred in this paper) and channel 0 (on the top down) indicates activity on the SPI bus. Numerical values are detailed in Table I.

1) $T_{\text{SPI}}$. SPI time lasts from the attempt to transmit the TXFIFO address to the end of the length field and MPDU (except Frame Check Sequence - FCS) transmission over the SPI bus. Some subsequent timings are related to the SPI bus too, but this is referred as $T_{\text{SPI}}$ since is the most dependent and lasting from any other using the SPI. It can be broken into three:

   a) $T_{\text{SPI-addr}}$: SPI access delay, TXFIFO address transmission and waiting time prior to data transmission.
   b) $T_{\text{SPI-MPDU}}$: Transmission of the length field and MPDU - except FCS. Data are sent in 10-byte blocks ($T_{\text{SPI-Blocks}}$ 92.75 $\mu$s) with a delay between blocks ($T_{\text{SPI-Dblocks}}$) used by the TinyOS scheduler to switch to other commands/tasks. All delays have the same length (33.25 $\mu$s) except the first (25.46 $\mu$s). In the event that the last block be shorter than 10 bytes, it can be computed based on the transmission of one byte and the separation between

![Fig. 1. Empirical entropy and average values on micaZ for 255 seeds](image-url)
bytes \((T_{SPI_{addr}}, 10.17 \mu s)\). Computing this time based on blocks and bytes rather than just bytes allows us to reach more accurate figures.

- **1) tSPI\_addr**: Release time of the SPI bus.
- **2) tSPI\_release**: From the SPI release to the beginning of the backoff there is just a major event: the backoff calculation. Such time is fixed, but not negligible (226.9 \(\mu s\)).
- **3) tbackoff**: Initial backoff as described previously.
- **4) tCCA\_SW**: CCA pin is checked right after the backoff, but a fixed additional software delay is intentionally inserted over the execution before sending the STXONCCA strobe.
- **5) tSTX\_SHR**: An STXONCCA strobe is issued over the SPI and if CCA by the transceiver returns as idle, the transmission starts after a \(t_{turn}\) time.

- **6) tMPDU**: Transmission of the length field and MPDU - now with FCS computed by the transceiver.
- **7) ttx2rx**: Transition from transmission to reception mode so that the transceiver is able to start another operation such as receive or send subsequent packets. The last falling edge in Fig. 2 sets the beginning of this transition while the subsequent rising edge corresponds to switching to reception mode.

### IV. Scenario

The scenario modeled in the current paper consists of a number of transmitting nodes and a sink which are within each other’s transmission range and there is not trigger message as in other papers [3]. A particular case of such model is a continuous transmission from one source node to a sink. This latter case allows us to estimate the upper bound throughput which may be useful as a bound indicator for researchers and practitioners even if their model does not match exactly this one or as in [1]. However, users seeking more complex scenarios can base their model on the information and considerations provided in section III.

### V. Analysis

#### A. Theoretical

From [17] we have (8) and (12), \(\alpha\) is the probability that the channel is busy at the CCA and is assumed to be constant regardless of the backoff length. The number of nodes that make up the scenario (within each other’s transmission range) is referred as \(n\). \(P_{loss}\) and \(\Gamma\) are the packet loss probability and number of packets served in a busy period -a period with channel activity- respectively. Ref. [8] defines both the CCA detection time \((T_{CCA})\) and the turnaround time \((T_{turn})\) to be equal to 128 \(\mu s\). \(T_{tx}\) denotes the transmission time of the PHY protocol data unit (PPDU). In the expectation of Head of Line Delay (12), the first summation represents the \(M\) tries before discarding the packet over busy channel and the summation on \(W_t\) is the backoff average value. \(S\) is the normalized throughput -normalized channel usage- which depends on the packet length.

\[
\alpha = \frac{(n-1)(1-P_{loss})E[\Gamma](t_{CCA} + T_{tx} + t_{turn})}{\frac{1}{\Gamma} + E[\Gamma]E[D_{Hol}]} \tag{8}
\]
\[
\alpha_{m_2} = \frac{(n-1)(1-P_{\text{loss}})E[\Gamma](t_{\text{CCA,SW}} + t_{\text{STX}} + t_{\text{turn}} + t_{\text{SHR}} + T_{\text{MPDU}})}{E[\Gamma]E[D_{\text{Hol}}]} + \frac{1}{E[\Gamma]E[D_{\text{Hol}}]}
\]

\[
E[D_{\text{Hol-Short}}] = T_{\text{SPI}} + (1 - \alpha_{m_2})(t_{\text{randC}} + E[T_{\text{backoff}}] + t_{\text{CCA,SW}} + t_{\text{STX}}) + \\
+ \sum_{i=1}^{M} \alpha_{m_2} \{(1 - \alpha_{m_2})(t_{\text{randC}} + E[T_{\text{backoff}}] + t_{\text{CCA,SW}} + t_{\text{STX}})\}
\]

\[
E[D_{\text{Hol}}] = \begin{cases} 
E[D_{\text{Hol-Short}}], & \text{if } \frac{1}{\alpha} < E[D_{\text{Hol-Short}}] \\
E[D_{\text{Hol-Short}}] + t_{tx2rx}, & \text{if } \frac{1}{\alpha} \geq E[D_{\text{Hol-Short}}]
\end{cases}
\]

\[
E[D_{\text{Hol}}] = \sum_{i=0}^{M} \alpha^{i}(1 - \alpha)\left\{ \sum_{i=0}^{v} \frac{W_i - 1}{2} \sigma + (v + 1)T_{\text{CCA}} \right\} + \alpha^{M+1} \left\{ \sum_{i=0}^{M} \frac{W_i - 1}{2} \sigma + (M + 1)T_{\text{CCA}} \right\}
\]

\[
S(l) = \frac{n(1-P_{\text{loss}})E[\Gamma]T_{\text{TX}}(l)}{E[\Gamma]E[D_{\text{Hol}}] + T_{\text{TX}}(l) + t_{\text{turn}}}
\]

\[
S_{\text{PHY}}(l) = \frac{n(1-P_{\text{loss}})E[\Gamma](t_{\text{SHR}} + T_{\text{MPDU}}(l))}{E[\Gamma]E[D_{\text{Hol}}] + t_{\text{SHR}} + T_{\text{MPDU}}(l) + t_{\text{turn}}}
\]

\[
S_{\text{AP}}(l) = \frac{n(1-P_{\text{loss}})E[\Gamma]T_{\text{MSDU}}(l)}{E[\Gamma]E[D_{\text{Hol}}] + t_{\text{SHR}} + T_{\text{MPDU}}(l) + t_{\text{turn}}}
\]

### VI. Comparison

Real measurements were performed to check the model described above. Such experiment along with the model itself provides the upper bound throughput of any peer to peer link for micaZ and consequently becomes meaningful as a first overview on the maximum throughput achievable on a multipoint to point link. For the experiment, nodes were 1 meter away, on 1.2 m. height (for 0.6 results were the same), the transmitter was set to 0 dBm and every payload was tested over 5 s. Since equations depend on \( \Gamma \) or its expectation, such value was taken in every comparison from the real measurements (tests) and therefore it is equal across the different curves of every figure. The term \( \frac{1}{\alpha} \) becomes negligible compared to the second term regardless the payload length -the experiment is nearly testing a steady state of the channel- therefore is not computed in Fig.3 and Fig.4. Exponent backoff for the theoretical model was set to 5 in order to tune it to micaZ settings as explained in section III. Consequently, no assumptions are made on the traffic model. The frame structure used is called I-Frame without TINYOS IP field in TinyOS 2.1 and is compliant with the standard; its MAC header is 11 bytes long. In order to compute \( S_{\text{PHY}} \) and \( S_{\text{AP}} \), \( P_{\text{loss}} \) was supposed to equal 0 as there was just one transmitting node. Results have proven such assumptions to be correct as the log on the receiver node reported no losses of the transmitted packets – every packet had a meaningful payload to keep a count.

Fig.3 shows the PHY throughput which gives us an insight of the channel usage. For the shortest PPDU (20 bytes) the channel usage is as low as 9.14% which is nearly the same as the the theoretical one - 10.81%. However, such difference becomes wider as PPDU length rises to 133 bytes: micaZ model is 35.36% while the theoretical one 44.63%. Fig.4 shows the AP throughput, which keeps to the same trend as the PHY throughput, but including the effect of headers and footers over the different payload lengths. It is useful for any practitioner regarding the application to be run on the motes. Theoretical and micaZ models in both figures are not simply offset, but their slopes are different also as they provide different information. AP throughput ranges from 0.45 to 30.31% due to the impact of overheads on the actual (application) channel usage. Such variation is slightly shorter than in PHY throughput.

As it can be seen in both figures, experimental results match micaZ model very close.

### VII. Conclusion

A real model of IEEE 802.15.4 in beaconless mode based on micaZ has been presented along with the reasoning involved. Compared to other models found in the literature, this contribution approaches the real world rather than making an interpretation of the standard as most of the contributions have made so far as referred in section II. Hence the current work analyzes the operation of the de-facto-standard
equipment used in research pointing out a different performance to what one might expect according to other works. Experiments provided may help any researcher or practitioner to model micaz (multi-point to point) has been computed and evaluated with real motes and at the same time the overhead impact on the communication is outlined.

REFERENCES


