ACM MobiCom 2012 Poster: Low-Cost Interferer Detection and Classification using TelosB Sensor Motes

Bastian Bloessl
bloessl@ccs-labs.org

Stefan Joerer
joerer@ccs-labs.org

Fabian Mauroner
fabian.mauroner@student.uibk.ac.at

Falko Dressler
dressler@ccs-labs.org

Computer and Communication Systems, Institute of Computer Science
University of Innsbruck, Austria

Inter-protocol interference is one of the most critical issues in wireless communication. For example, this becomes extremely problematic in environments where robustness and real-time communication need to be considered, e.g., in industrial automation or health care applications. Recently, possible approaches for interference mitigation have been described in the literature assuming that the interferer is known in advance. We contribute to this line of research with a framework for interferer detection and classification. Essentially, we use a simple IEEE 802.15.4 transceiver as for example used on the TelosB sensor motes to scan the 2.4 GHz ISM band. This band is used by different technologies including Bluetooth, WiFi, and cordless phones. The key challenge is the accurate timing of the scanning of the frequency band. The presented framework supports flexible descriptions of such scan jobs allowing to adapt to the detectors requirements, depending on the interfering protocols.

I. Introduction

We investigate the possibilities for low-cost interferer detection and classification based on a rather cheap and IEEE 802.15.4 standard conform transceiver chip integrated with the TelosB sensor mote platform. Inter-protocol interference is becoming a predominant problem for many wireless networks including WiFi access to the Internet. This is because the unlicensed ISM bands, especially the 2.4 GHz band, is a source of increasing interference among the different protocols. For example, WiFi, ZigBee-based sensor networks, Bluetooth devices, and dedicated application specific protocols have been developed. Studies have shown that insufficient knowledge of interfering signals may lead to substantial performance degradation [7,9]. However, current protocols, e.g., IEEE 802.11 or IEEE 802.15.4, do not take inter-protocol interference into account. This reduces the robustness, which is critical if reliable communication is required, e.g., in industrial automation environments [3]. Adaptive channel switching strategies can be realized assuming better knowledge about the interferer [5].

In 2009, Chowdhury and Akyildiz studied the problem of interferer detection and formulated a general methodology [4]. In particular, the authors used sensor nodes to collect spectral data which was used for offline interferer identification. Most of the commercial spectrum analyzers, e.g., AirMagnet Spectrum XT, Agilent Spectrum Analyzer or Bandspeed AirMaestro employ specialized hardware. They are expensive and cannot be easily integrated into an interference-aware protocol design. However, tools such as Wi-Spy and Ubertooth are more affordable, but provide limited capabilities. Based on commodity hardware, Airshark [8] uses WiFi cards to identify different interference sources.

We present an extended version of Spectrum Analysis Framework for Interferer Classification (SaFIC) [2], a sensor mote based spectrum analysis framework for the 2.4 GHz ISM band. Our framework allows to measure the Received Signal Strength Indicator (RSSI) in a predefined spectrum and visually displays the signal strengths and their corresponding frequencies in real-time. It is implemented on typical sensor mote hardware using a Chipcon CC2420 transceiver chip that is widely used in many application domains including health care, building automation, and industry automation. Our extended architecture supports fine grained scan jobs with very precise timing control to accurately detect multiple interfer-

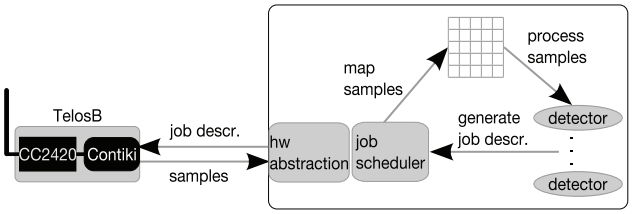

Figure 1: Interferer detection architecture

ing networks at the same time. For improved scan performance, the jobs can be organized that only frequencies under investigation need to be evaluated. For rapid prototyping of detector modules, we support a flexible language to define complex scan jobs.

II. Framework Architecture

An overview of the framework is depicted in Figure 1. It consists of a host application and a customized firmware that is running on a TelosB mote [2, 6]. The TelosB uses the very common Chipcon CC2420 transceiver chip. The transceiver mainly determines the measurement speed and resolution and hence defines the limitations of our framework.

II.A. TelosB Firmware

The firmware is based on the Contiki operation system. It receives and parses the scan job descriptions which define how measurements should be recorded. The format and the capabilities of these jobs are described in Section II.C. Before a measurement is started, all watchdogs and interrupts of the mote need to be disabled. This ensures that the measurements are not delayed or interrupted by any other event that may occur. Thus, the measurement process becomes deterministic and the results can be interpreted as the relative timing between the measurements is known. When the job is completed the watchdogs and interrupts are enabled again and the recorded samples are transmitted to the host computer for further processing.

II.B. Host Component

The host part of our framework is shown on the right hand side of Figure 1 which in essence consists of the job scheduler, multiple detection modules and a global matrix with measurement results. The job scheduler queries to detection modules for jobs in a round-robin fashion and transfers them to the sensor mote. When a sampling job has been completed by the sensor mote, the job scheduler receives the measurements and stores them in the matrix. Thereafter, the detection modules process the updated samples. Additionally, the matrix is visualized live in order to support an immediate, intuitive understanding of the issued scan jobs and the detection algorithms. Thus, the limitations of the sensing device can be investigated quickly and new findings can be incorporated in the development of detector modules.

II.C. Job Description Language

We extended the possibilities to describe jobs substantially. While the initial version supported only the most basic description of a single measurement, we can now nearly arbitrary combine scans, delays, and loops. This enhanced description of jobs can be formally specified in Backus-Naur Form as

⟨*job*⟩ ::= ⟨*command*⟩ `'end'`

⟨*command*⟩ ::= ⟨*scan*⟩ | ⟨*loop*⟩ | ⟨*delay*⟩
| ⟨*command*⟩

⟨*scan*⟩ ::= `'scan' START END STEP DWELLS SWEEPS`

⟨*loop*⟩ ::= `'loops' REP` ⟨*command*⟩ `'loope'`

⟨*delay*⟩ ::= `'delay' DELAY-TIME .`

The *measurements* define where samples are to be recorded in the frequency band. In addition to the `START` and `END` of the spectrum also a `STEP` width can be specified. Due to the limitations of the transceiver, we can only measure frequencies that are multiples of 1 MHz and in the range from 2.4 GHz to 2.5 GHz. The `DWELL` and `SWEEP` parameters have the same semantic as in [2] and define how many samples are taken on a certain frequency and how often we hop through the spectrum. The arguments of *loop* and *delay* (`REP` and `DELAY-TIME`) specify how often the *commands* inside the loop are repeated and the duration to wait respectively.

III. Framework Capabilities

SaFIC offers the following capabilities for developing interferer detection and classification modules: **Wide range of sampling patterns**, i.e., the scan parameters offer extensive possibilities for sampling the entire frequency range; **Interleaving of scans**, i.e., the possibility to have two independent scans within one job giving the user the possibility to sample multiple channels nearly in parallel; **Delays**, i.e., support for delay commands between single measurements,

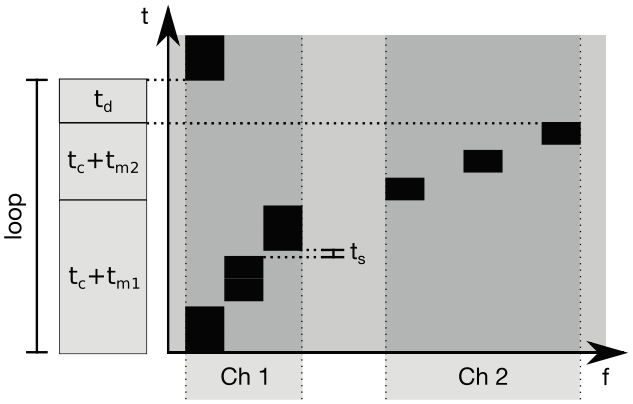

Figure 2: An interleaved measurement

which might be used to check for specific timings during protocol classification.

An example job is depicted in Listing 1 and Figure 2. It uses two specific scans to measure the signal spectrum of two frequency ranges *Ch 1* and *Ch 2*. The first scan uses two `DWELLS` in one `SWEEP`, whereas the second scan uses one `DWELL` in a single `SWEEP`, thus, representing two different sampling strategies. The `STEP` in the second scan defines a frequency stepping of 2 MHz. Finally, Listing 1 introduces a delay (t_d) after the two scans have been recorded.

```
loops 2
    scan f1 f2 1 2 1     // Ch1
    scan f3 f4 2 1 1     // Ch2
    delay t_d
loope
```

Listing 1: Example job

III.A. Timing of the Measurements on TelosB

In order to interpret the measured RSSI values, it is crucial that the relative timing between the samples is known. As we stated in Section II.A interrupts are disabled during the measurement process and, thus, its behavior becomes deterministic. Based on timing measurements and analysis of the assembler output, we created a model for our job descriptions. All measurements were made with the help of an oscilloscope that we connected to an I/O pin of the mote. According to our findings every *command* adds a constant delay $t_c = 7\,\mu s$. For a *scan* we figured out two relevant timings, while one covers a single measurement which takes a time $t_r = 80\,\mu s$, switching the frequency adds a delay $t_s = 300\,\mu s$. Apart from additional instructions that are required to tune to another frequency, t_s also includes the time to generate the first RSSI value. According to the standard, each RSSI measurement

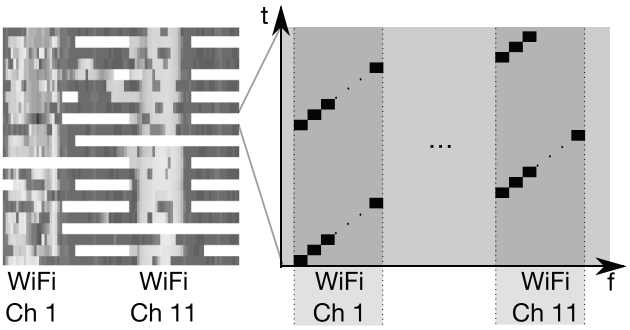

Figure 3: Example WiFi detection

should be averaged over eight symbols [1], which correspond to 128 µs. Thus, with the number of sampled frequencies n_f, the total time for a *scan* t_{scan} can be calculated as

$$n_f = \lfloor (END - START)/STEPS \rfloor$$
$$t_{scan} = n_f \cdot SWEEP \cdot DWELL \cdot t_r + n_f \cdot SWEEP \cdot t_s + t_c.$$

With this formula the timings t_{m1} and t_{m2} from Figure 2 can be determined. The delay can be specified in the range from 7 µs to 50 ms. Currently it is implemented as a busy-waiting loop that needs three CPU cycles per iteration. Hence, the delay resolution is 0.77 µs which is determined by the clock speed of the microcontroller that runs at 3.9 MHz.

III.B. WiFi Detection

To demonstrate the features of our framework we implemented a simple detection module for WiFi. Figure 3 shows a screenshot of the visualization while the WiFi detector is running in our office. It can be seen that two separate WiFi networks have been recognized. To accomplish this, the detector takes rough sampling data from the entire 2.4 GHz spectrum and matches it with the spectral masks that are defined in the standard. Based on this matching a threshold is used to decide where WiFi networks may be present. Then, the detector focuses on the two parts of the spectrum where these networks operate and samples these candidate channels in an interleaved manner as shown on the right side of Figure 3. With this approach, the module is able to detect individual channels as long no other protocol interferes. However, differentiating multiple interference sources that operate in the same frequency range remains as open challenge.

IV. Conclusion and Future Work

We extended our interferer detection framework SaFIC to support flexible description of measurement jobs. For this reason, we defined a job description language which offers arbitrary combinations of loop, delay, and multiple scans within one measurement job.

The main advantage of our new framework is to support interleaved measurements. As a proof of concept, we demonstrated a WiFi detector that is now able to identify two WiFi networks operating on different channels more accurately. Furthermore, we improved the precision of the timings of each measurement job.

In future work, we aim to develop optimized detector modules that, for example, make use of the delay command to schedule local preprocessing tasks on the sensor mote. This would allow to transfer most of the detector logic to the mote as well.

References

[1] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). IEEE Standard 802.15.4-2006, IEEE, 2006.

[2] B. Bloessl, S. Joerer, N. Nordin, C. Sommer, and F. Dressler. SaFIC: A Spectrum Analysis Framework for Interferer Classification in the 2.4 GHz Band. In *IEEE INFOCOM 2012, Demo Session*, Orlando, FL, March 2012. IEEE.

[3] F. Chen, R. German, and F. Dressler. Towards IEEE 802.15.4e: A Study of Performance Aspects. In *IEEE PERCOM 2010, IQ2S Workshop*, pages 68–73, Mannheim, Germany, March 2010.

[4] K. R. Chowdhury and I. F. Akyildiz. Interferer Classification, Channel Selection and Transmission Adaptation for Wireless Sensor Networks. In *IEEE ICC 2009*, pages 1–5, Dresden, Germany, June 2009.

[5] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and Mitigating the Impact of RF Interference on 802.11 Networks. In *ACM SIGCOMM 2007*, pages 385–396, Kyoto, Japan, August 2007.

[6] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *ACM/IEEE IPSN 2005*, pages 364–369, Los Angeles, CA, April 2005.

[7] S. Shin, H. Park, and W. Kwon. Mutual Interference Analysis of IEEE 802.15.4 and IEEE 802.11b. *Elsevier Computer Networks*, 51(12):3338–3353, 2007.

[8] R. Shravan, P. Ashish, and B. Suman. Airshark: Detecting Non-WiFi RF Devices Using Commodity WiFi Hardware. In *ACM SIGCOMM 2011*, pages 2–4, Berlin, Germany, November 2011.

[9] A. Sikora and V. Groza. Coexistence of IEEE 802.15.4 with Other Systems in the 2.4 GHz-ISM-Band. In *IEEE IMTC 2005*, volume 3, pages 1786–1791, Ottawa, Canada, May 2005.