IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 30, 2019, accepted May 17, 2019, date of publication May 22, 2019, date of current version June 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918382

In-Band Controllable Radio Interference Generation for Wireless Sensor Networks

EMANUELE LATTANZI AND VALERIO FRESCHI
Department of Pure and Applied Sciences, University of Urbino, 61029 Urbino, Italy
Corresponding author: Emanuele Lattanzi (emanuele.lattanzi@uniurb.it)

This work was supported by the Department of Pure and Applied Sciences (DiSPeA), University of Urbino, under Institutional Grant 2018.

ABSTRACT Interference signals negatively impact the performance of wireless embedded systems. The increased packet losses, delays, and energy consumption experienced by devices operating in environments subject to interference are particularly critical in constrained systems such as wireless sensor networks. The need to design and test systems for mitigating the effects of interference prompts for the capability of reproducing in a controllable way suitable interference signals. Solutions have been proposed recently, which tackle the problem by making use of 802.15.4 compliant radio transceivers, like those available on board of commonly used sensor nodes, thus paving the way for low cost and repeatable generation of interference which could reliably emulate real-world scenarios, for instance densely deployed networks. In this paper, we present an investigation regarding the emulation of interference sources by means of 802.15.4 radios on novel 32-bit wireless system-on-chip. The study is based on an extensive experimental evaluation, providing novel insights into the main features of the system. In particular, the effects of interference on the communication link, measured in terms of packet reception rate, are investigated for different parameters (namely, duty cycle and power of the interference signal, communication protocols, and payload size), and results are discussed concerning the feasibility of emulating background noise by means of the analyzed techniques.

INDEX TERMS Interference generation, wireless sensor networks, performance evaluation.

I. INTRODUCTION

The increasing diffusion of devices that exploit the 2.4 GHz ISM band for communication purposes has progressively led to saturation of the available radio-frequency spectrum resources. As a consequence, interference between signals sharing the same physical channel is an issue that growingly affects wireless systems and novel solutions need to be studied in order to cope with its negative impact on the overall quality of communication [1], [2]. Dealing with interference is particularly problematic in the context of Wireless Sensor Networks (WSNs), which are characterized by severe constraints in terms of computation, communication and energy requirements [3]. Indeed, interference results into reduced reliability because of higher packet loss levels and, in turn, into higher latency and energy expenditure [4]. Needless to say, the decrease of dependability of WSN systems operating in environments subject to interference is crucial in safety critical applications, for instance in some healthcare or industrial settings.

Hence, there is a strong motivation to research solutions for counteracting or preventing the effects of interference on wirelessly networked embedded systems. While simulation represents a valid choice for a first experimental assessment of a new technique, experiments on real-world hardware provide further valuable insights which are often essential for a comprehensive evaluation.

However, testing novel techniques with an adequate level of accuracy with respect to realistic scenarios in a reproducible way, is itself an issue which requires careful development and characterization of interference generation systems. Consequently, some authors have recently started to investigate the design of systems for realistic and reliable generation of interference patterns [5], [6].

According to a widely adopted taxonomy, interference can be usually classified into *out-of-band* (or external) and *in-band* (or internal). The former refers to spurious electromagnetic radiation that is produced by consumer electronics devices operating in different frequency bands, that

The associate editor coordinating the review of this manuscript and approving it for publication was Ghufran Ahmed.

partially overlap the communication band. The latter refers to electromagnetic signals generated in a frequency range that completely overlaps the communication band, such those generated in a network where concurrent transmissions may occur and collide at a given receiver side.

Boano *et al.* presented a controlled generation of out-of-band interference by means of a two-step process consisting of a collection phase (during which real interference traces are detected and recorded) and by a replicate phase (during which a wireless transceiver is used for replaying the recorded signals) [6].

In-band interference can be in principle generated by programming a transceiver to continuously transmit broadcast packets at a given rate in order to produce a packet storm that increases, to a given extent, latency and collisions. This solution, despite being a simple one, presents the drawback of a poor tuning capability and of a remarkable dependency from the software stack of the transmitting radio transceiver. In [5], [6] the authors proposed a method to generate a controlled level of interference in the 2.4 GHz ISM band by means of 802.15.4-compliant sensor devices. In particular, they introduced a solution that exploits the CC2420 radio chip to obtain adjustable interference functions and patterns, noise levels and packet loss rates. The resulting methodology provides a cheap and simple, yet effective solution for the generation of interference signals, which represent the basis for our research study.

Specifically, in this work we aim at providing a further characterization of the above mentioned methodology by means of the following contributions:

- we present the implementation of interference generation methods on a new, up to date, hardware platform, namely a 32bit wireless system-on-chip;
- we describe a detailed evaluation and characterization of the system through an extensive set of experiments;
- we investigate and discuss the effects of generated interference on communication performance, with emphasis on the effect of given parameters (e.g. the duty cycle of the interferent signal) on given metrics (e.g. the packet reception rate and the received signal strength) and on the emergence of some characteristic features (e.g. a steep phase transition in packet reception rate for background noise emulation).

The presented study provides therefore a thorough investigation which could support WSN designers in assessing the performance of novel techniques and systems operating under varying interference conditions.

The remainder of the article is organized as follows: in Section III we describe the techniques adopted for interference generation, in particular the methods for emulating a *packet storm* (together with a mathematical characterization of the communication channel under this type of interference) and a *background noise* floor; in Section II we examine related contributions in state-of-the-art literature; in Section IV we introduce the system set-up used for the experimental evaluation; in Section V we describe and discuss the outcomes of the extensive set of experiments conducted; in Section VI we recapitulate the main contributions of the article and set forth some final considerations.

II. RELATED WORK

Many works have investigated the role of interference on communication in wireless networks or, more specifically, in wireless sensor networks; this often entailed the artificial generation of given interference signals in a controlled manner for testing purposes. However, few researches have truly focused on the issue of designing reliable, realistic, tunable, and affordable systems capable of emulating interference sources. As a consequence, with some notable exceptions (e.g. [5], [6]), most of the related works are limited to a brief description of the particular system chosen for performing some experimental activity, instead of discussing the features that are involved in exploiting one of the available methods or instead of proposing a novel one.

Bertocco *et al.* adopted a signal generator coupled with a log periodic antenna for studying interference in industrial applications. This method represents a flexible solution but incurs higher costs with respect to the solution investigated in our work because of the need of supplemental hardware.

Packet storm has been another investigated alternative, which consists of exploiting a 802.15.4-compliant transmitter (a jammer node) for sending broadcast packets at a specified rate in order to increase packet congestion and delay in the network [7]–[11]. As pointed out in [5], this method is hindered by the difficulty of achieving precise control over the performance of the system. For instance the software stack overhead impairs the continuity of the generated interference, or it requires complex synchronization between the jammer node and the transmitter node).

More recently, the adoption of Universal Software Radio Peripherals (USRP) dedicated devices [12] made it possible to improve the accuracy of the interference generation systems to higher levels and has been adopted as a possible solution for testing interference-aware power control mechanisms [13], [14]. However, they currently represent a too costly solution because of the need of specialized hardware, which makes it difficult an adoption on large scale testbeds. Indeed, in dense and broad (from a spatial point of view) deployments, the capability of reproducing interference patterns at different locations at reduced costs could represent a useful (whenever not mandatory) feature.

III. GENERATING IN-BAND INTERFERENCE

A practical way to generate in-band interference has been described by Boano et al. in 2009 [5]. In particular they discuss the in-band interference generation by means of some RF transceivers available on sensor nodes, such as the Texas Instruments' CC2420. In fact, thanks to the different transmit test modes of these RF-chips it is possible, without the need of any other hardware device, to send a continuous carrier signal at a predefined frequency which can emulate a background noise. Furthermore, the carrier signal can also

```
1   /* reset tune frequency calibration */
2   REG(RFCORE_XREG_FSCAL1) = 0x00;
3   /* set automatic gain control target value */
4   REG(RFCORE_XREG_AGCCTRL1) = 0x15;
5   /* set TX anti-aliasing filter bandwidth */
6   REG(RFCORE_XREG_TXFILTCFG) = 0x09;
7   /* set the multiplier in frequency calibration */
8   REG(RFCORE_XREG_FSCAL1 = 0x01;
9   /* set random data from hardware generator */
10  REG(RFCORE_XREG_FRMCTRL0) = 0x43;
11  /* reset frame handling */
12  REG(RFCORE_XREG_FRMCTRL1) = 0x00;
13
14  /* set communication channel */
15  cc2538_rf_channel_set(CC2538_RF_CHANNEL);
16
17 #if MODULATED
18  /* enable DAC output from random data */
19  REG(RFCORE_XREG_MDMTEST1) = 0x08;
20 #else
21  /* transmit a constant tone */
22  REG(RFCORE_XREG_MDMTEST1) = 0x18;
23 #endif
```

FIGURE 1. C code of the Contiki interferer initialization function.

be continuously modulated using the internally available random number generator to reproduce the transmission of data packets. Using this radio chip Boano et al. presents two alternative strategies to obtain a tunable and reproducible interference. The first one called *packet storm emulation* is aimed at producing an intermittent randomly-modulated signal consisting of a square wave that can easily emulate a storm of data packets. The second one, on the other hand, is aimed at manipulating the SNR of the communication channel by imposing a continuous unmodulated carrier. This strategy called *background noise emulation* can easily be tuned by varying the transmission power of the radio chip at a desired level. In this work we implement, test and characterize these two strategies on the novel Texas Instruments' CC25xx family of system-on-chip radio transceiver [15], to provide a thorough method for generating repeatable and tunable interference patterns for WSNs experiments. In particular, a VirtualSense node equipped with the new Texas Instruments CC2538 and with an instance of Contiki operating system has been used as interference generator.

Despite this new chip family derives from the CC2420 they show no register compatibility so that the configuration routines have been rewritten ad-hoc. In order to clarify some programming practice aspects, and also to better support research reproducibility, we report and discuss a few key points of the implemented code.

For instance, Figure 1 shows the Contiki initialization code used to configure the RF chip into test modes.

Rows 1 to 8 contain standard code recommended by Texas Instruments to reset and calibrate the frequency generator while in row 10 we configure the random number generator used later to modulate the transmitted tone. Finally, rows 19 and 22 enable respectively the randomly-modulated and un-modulated test modes. Notice that, in unmodulated test mode, the chip transmits a constant tone centered on the frequency defined by the selected channel which shows a narrow power spectrum while the randomly-modulated signal distributes the power spectrum across the channel

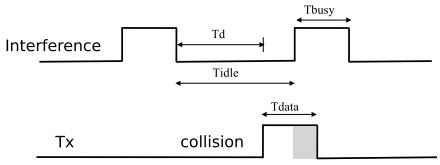

FIGURE 2. Periodic interference square wave timing.

bandwidth. In the modified radio driver, built for this this work, the modulated and unmodulated option can be independently chosen from the type of the generated interference (i.e. packet storm or background noise).

A. PACKET STORM EMULATION

The packet storm emulation tries to emulate the in-band interference that may be generated in a dense WSN where the number of control and data messages can be very high due to the large number of installed nodes. In a real dense deployment this kind of interference depends on four independent variables: the node transmission power, the packet length, the elapsed time between each packets, and the distance between nodes. The combination of these parameters leads to a plethora of possible interfering conditions which can be hardly characterized and which are far from being controllable and tunable. The packet storm emulation presented by Boano et al., [5] which is exploited in this work, reduces the degree of freedom without compromising the realism of the effects of the interference on the nodes communication. In particular, the interfering node (*interferer*) has been programmed to generate a square wave characterized by two parameter which are: the time in which the transmitter is on (T_{busy}) and the time in which it is off (T_{idle}). According to this definition the interferer can by characterized by its channel occupancy rate ρ, also known as the duty cycle of the square wave, by means of Equation 1:

$$\rho = \frac{T_{busy}}{T_{busy} + T_{idle}} \tag{1}$$

In order to further reduce the number of independent variables the transmission power of the interferer has been configured to its maximum level (i.e. 7dBm for CC2538) so as to avoid any kind of communication during the busy time. In this way it is possible to tune the amount of produced interference by simply modify the ratio between T_{busy} and T_{idle} time in order to achieve any desirable effect (in terms of packet corruption rate for instance) in the communication link. Of course, because the 802.15.4 standard supports a data transfer rate of 250 kbs, which implies a transmission time of about 4 ms for the largest available packet (127 bytes), it is important not to fall below this value when setting the T_{idle} in order to not completely avoid packet transmission.

According to this definition the interfering signal can be represented as a binary value changing over the time where during the "busy" level any communication is prevented because of the data corruption.

```
1  PROCESS_THREAD(cc2538_rf_interferer_process, ev, data)
2  {
3    rtimer_clock_t t0;
4    PROCESS_BEGIN();
5
6    PRINTF("Starting RF-interferer main loop\n");
7
8    /* set transmit power */
9
10 #if PACKET_STORM
11   cc2538_rf_power_set(CC2538_RF_MAX_POWER);
12 #elif BACKGROUND_NOISE
13   cc2538_rf_power_set(CC2538_RF_TX_POWER);
14   CC2538_RF_CSP_ISTXON();
15 #endif
16
17   while(1){
18
19 #if PACKET_STORM
20       /* send the ISTXON opcode *
21        * to immediately enable TX */
22       CC2538_RF_CSP_ISTXON();
23
24       /* actively wait for T_BUSY milliseconds */
25       t0 = RTIMER_NOW();
26       while(RTIMER_CLOCK_LT(RTIMER_NOW(), t0+T_BUSY)){}
27
28       /* send the ISROFF opcode *
29        * to immediately disable TX */
30       CC2538_RF_CSP_ISRFOFF();
31
32       /* actively wait for T_IDLE milliseconds */
33       t0 = RTIMER_NOW();
34       while(RTIMER_CLOCK_LT(RTIMER_NOW(), t0+T_IDLE)){}
35 #elif BACKGROUND_NOISE
36       __asm(" nop");
37 #endif
38   }
39   PROCESS_END();
40 }
```

FIGURE 3. C code of the Contiki interferer process.

In particular, as represented in Figure 2, given an idle interval of length T_{idle} and given a data packet transmission time T_{data} a collision may occur if a data packet is transmitted after a time interval T_d for which $(T_{data} + T_d > T_{idle})$. Notice that, in this definition each colliding bit leads to the whole packet corruption which will be discarded at physical level. Despite the approximation, this assumption still appears realistic in WSN thanks to the widespread lack of any Error Correction Code (ECC) strategies at the receiver level which could, in some cases, compensate for the corruption of a packet.

Figure 3 reports a snapshot of the C code containing the Contiki process built to manage the interferer. In particular, it consists of a single Contiki process which executes an infinite loop (rows from 17 to 38) in charge of repeatedly turning on and off the transmission according to two parameters `T_BUSY` and `T_IDLE` representing, respectively, T_{busy} and T_{idle}. Before starting the main loop, the transmission power of the radio chip is configured at its maximum level for the packet storm interferer (rows 10-11), or to a predefined value in case of the background noise emulator (rows 12-15). In the packet storm main loop the radio transmitter is switched on by sending the related command (row 22) and then it is kept active until a specific amount of time (defined by `T_BUSY`) has elapsed (rows 25-26). Then the transmitter is switched off (row 30) for `T_IDLE` msec (rows 33-34). On the other hand, if the interferer is configured to emulate the background noise, since the interfering signal must be continuous, the main loop body must never change the radio state so that it is reduced to a dummy instruction (row 36).

Mathematical Model: We now introduce a mathematical model aimed at evaluating the impact of an in-band interference process on the reliability of the communication link, which can be used to predict the effect of the packet storm emulation process, that we used as a comparison term for experimental assessment.

Assuming a system composed of:

- an interference source emulating a packet storm (as described in III-A);
- a receiver always listening the channel for incoming transmission;

following the approach described in [16], [17]), we may write the probability of transmission failure as follows:

$$p_f(L, SNR) = 1 - (1 - p_c(L)) \cdot (1 - p_s(L, SNR)) \quad (2)$$

where $p_c(L)$ represents the probability of transmission failure imputable to packet collision, while $p_s(L, SNR)$ is the probability of transmission failure due to low Signal-to-Noise ratio (SNR for short), which can be derived from the following:

$$p_s(L, SNR) = 1 - (1 - BER(SNR))^{L_p+L_m+L} \quad (3)$$

In Equation 3, L is the payload length, L_p is the overhead of the physical layer and L_m that of the MAC layer.

A common abstraction of a communication channel subject to interference entails the adoption of a semi-Markov model with two states corresponding to idle and busy channel [17]. Each state can be fully specified in terms of its probability density function. Let the mean duration length of the idle state be τ_{idle}, and the mean duration length of the busy state τ_{busy}. We can compute the channel occupancy rate ρ as the duty cycle of the square wave signal that emulates the burst of packets as specified in Equation 4:

$$\rho = \frac{\tau_{busy}}{\tau_{busy} + \tau_{idle}} \quad (4)$$

The collision probability p_c can be computed as a function of two terms ($p_{c|busy}$ and $p_{c|idle}$) which represent, respectively, the probability of failure when the transmission started in presence or absence of interference:

$$p_c = \rho \cdot p_{c|busy} + (1 - \rho) \cdot p_{c|idle} \quad (5)$$

On the other hand, p_c can be expressed as a function of the collision probability of data packets (p_c^{data} and ACK packets (p_c^{ack}):

As well, p_c can also be written as:

$$p_c = 1 - (1 - p_c^{data}) \cdot (1 - p_c^{ack}) \quad (6)$$

In Equation 6 the second term can be neglected if the communication protocol considered doesn't entail any re-transmission mechanism (i.e. $p_c^{ack} = 0$) as for the case of our study (where we focused on broadcast data transmission). The collision probability of data packets can be in turn written as:

$$p_c^{data} = \rho + (1 - \rho) \cdot p_{c|idle}^{data} \quad (7)$$

Assuming that the idle state length duration amounts to a quantity T_{idle}, and that the time needed to transmit a data packet is T_{data}, a collision occurs if a data packet is transmitted ($T_{idle} - T_{data}$) seconds after the interferer has entered the idle state. Let T_d be the difference between the time at which a data packet is received and the time at which the interference enters the idle state (Figure 2). We have that $0 \leq T_d \leq T_{idle}$.

If the interference signal is represented by a periodic square wave of duty cycle ρ, we can assume the mean duration of the idle state to be $\tau_{idle} = T_{idle}$ and the mean duration of the busy state to be $\tau_{busy} = T_{busy}$. Given a transmission bit rate R_{br}, we can therefore derive $p^{data}_{c|idle}$ as:

$$p^{data}_{c|idle} = Pr[T_{idle} < T_d + T^{data}_{pkt}] = \frac{T_{data}}{T_{idle}} = \frac{L}{R_{br} \cdot \tau_{idle}} \quad (8)$$

From Equations 8and 7 we obtain:

$$p^{data}_{c} = \rho + (1 - \rho) \cdot \frac{L}{R_{br} \cdot \tau_{idle}} \quad (9)$$

The mathematical model that describes the reliability of the communication channel can be completed by taking into account the characteristics of the radio receiver. While several investigations can be conducted (depending on the type of receiver and of MAC protocol to be studied), we present in this work the results for the case of a communication system composed of: *i)* a transmitter for which we do not assume any capability of detecting the interference affecting the receiver (i.e. the transmitter doesn't implement any collision avoidance strategy; *)* a receiver that continuously listens to the wireless channel while waiting for possible packets (hereafter also denoted as *always on receiver*). While this type of communication pattern is critical for energy constrained settings (indeed low power WSNs make use of cycled receivers to alleviate power consumption), it represents a useful baseline for understanding the interplay between the interference source and the reliability of the communication link [16].

The probability of a broadcast transmission failure generated by a packet collision can be computed by taking p^{data}_{c} from Equation 9, resulting into:

$$p_c = \rho + (1 - \rho) \cdot \frac{L}{R_{br} \cdot \tau_{idle}} \quad (10)$$

It is clear from Equation 10 the dependence of p_c from the mean length and the channel occupancy rate of the idle state of the interferer, the transmitter bit rate and the length of the packet payload.

This value of collision probability can be finally used to estimate the reliability R of the system (transmitter sending broadcast packets and always on receiver with interference only at the receiver side):

$$R = 1 - p_c \quad (11)$$

B. BACKGROUND NOISE EMULATION

Emulating the background noise entails, after the first reset and calibration of the radio chip, to set-up the selected transmission power and then start transmitting a constant tone. The Texas Instruments' CC2538 system-on-chip provides several transmission power levels where the recommended configurations start from −13dBm up to +7dBm for a total amount of 13 operative levels. The choice of the transmission power can be taken during the initialization phase before enabling the transmission or, otherwise, it can be changed runtime with the only precaution to turn off and on again the transmission.

Finally, as described by Boano et al. the intuitive way to emulate a background noise is to transmit a constant tone on the center of the channel frequency without any kind of modulation. Despite this, in the implementation described in this work the possibility to generate both an unmodulated or a randomly modulated signal have been preserved in order to investigate a possible different effect on the communication performance.

IV. EXPERIMENTAL SET-UP

In this section we provide an extensive description of the experimental set-up conceived to characterise the performance of the proposed in-band interference generator. In particular we investigate the interferer effectiveness by measuring its effects, in term of reliability and in term of produced Received Signal Strength Indication (RSSI), on a communication link established between two WSN motes. The set-up was composed of: *i)* a sender node, continuously transmitting broadcast packets of a given, specified, length; *ii)* a receiver node listening to the channel for incoming transmissions; *iii)* an interference generator in charge of emitting a signal according to specific patterns. The sensor nodes and the interferer were positioned at the vertices of an equilateral triangle of side 20 *cm* for the entire duration of the experiments to remove the uncertainty on the path loss.

In order to measure the effects of the interferer without any distortion introduced by adaptive behaviors implemented in the network stack of the sensor nodes, we disabled the carrier sensing and collision avoidance (CSMA/CA) mechanism and any low-power radio duty cycle protocols thus allowing the communication only by means of the 802.15.4 standard physical layer. In fact, a sender implementing a CSMA/CA solution can sense an interfering signal and decide to postpone the transmission for a while thus reducing the interference effect on the ongoing transmission. Similarly, preamble sampling low-power MAC protocols can mitigate the entity of the interference by implementing proper retransmission policies.

Each experiment was conducted on the IEEE 802.15.4 channel 26 which does not overlap WiFi communications, so to remove any possible unwanted interference. Results and statistics were collected from experiments consisting of a transmission of about 1000 broadcast packets during a 30 minutes time frame. Inter-packet time was approximately 1.8 seconds with a small amount of jitter introduced to avoid any potential synchronization effects.

The RSSI generated by the interfering node was measured by means of a Texas Instruments' SmartRF06 evaluation board mounting a CC2538 evaluation module set

in continuous receive mode. The board was connected to a Windows machine running an instance of Texas Instruments' SmartRF Studio which was recording the RSSI value [18], [19].

A. WIRELESS SENSOR NODE

The sensor node chosen to build the transmitter, the receiver and the interferer is VirtualSense [20], an ultra low-power sensor platform based on the Texas Instruments' CC2538 system-on-chip designed for 2.4-GHz IEEE 802.15.4 applications [15]. VirtualSense is based on the Contiki operating system [21] hosting a Darjeeling Java compatible virtual machine [22]. Both of them have been properly modified in order to enable concurrent execution of Java tasks while maintaining a reduced power consumption (about few μW on average) [23]. VirtualSense inherits the low-power Contiki network stack called *Rime* implementing the following four software layers: *i) Network*; *ii) MAC* - Medium Access Control; *iii) RDC* - Radio Duty Cycling; *iv) Radio*. The Radio Duty Cycling and the Radio layers provide respectively power saving strategies, such as X-MAC, ContikiMAC, and LPP, and the software drivers needed to manage the RF chip [24], [25], [26].

In our experiments both the sender and the receiver run a Contiki OS configured with a `nullmac_driver` and a `nullrdc_driver` in order to remove the CSMA/CA mechanism and the radio duty cycling protocol. On the third VirtualSense node a modified radio layer, implementing the interference generation process described in section III, has been installed. Each node was powered at 3.3V through a NGMO2 Rohde & Schwarz dual-channel power supply [27] and monitored by means of a National Instruments NI-DAQmx PCI-6251 16-channel data acquisition board connected to a BNC-2120 shielded connector block [28], [29].

V. RESULTS

In this section, we present the results of extensive experiments aimed at evaluating the effectiveness of the proposed system. In particular, we want to estimate the effect of the interference on the communication performance of an IEEE 802.15.4 wireless link in terms of reliability. The reliability has been measured by taking into account the packet reception rate (PRR) of a broadcast communication subject to a predefined value of interference. A particular attention has also been paid to assess the possibility of tuning and to the reproducibility of the interfering conditions which lead to a given PRR value.

The characterisation of the interferer has been carried out by means of two different sets of experiments according to the two operating mode called *packet storm emulation* and *noise emulation*.

A. PACKET STORM

In this set of experiments the duty cycle of the interferer has been changed by varying the T_{idle} parameter while the value of T_{busy} was kept constant and equal to 4ms. Notice that a busy time of about 4ms, according to the 250Kbit/s bit rate of the 802.15.4 standard, corresponds to a transmission of a packet containing about 127 bytes which is the maximum available size. Each run has been repeated 5 times and at the end of each one the related PRR has been measured.

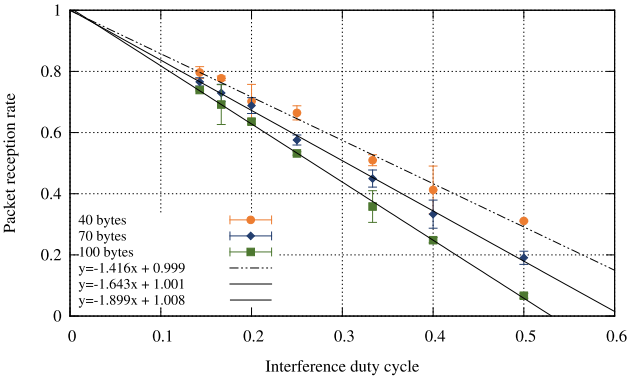

FIGURE 4. Packet reception rate when varying the interference duty cycle for different packet size.

Figure 4 shows the measured PRR, together with the calculated error bars, when the interferer duty cycle increases from about 0.14 (corresponding to $T_{idle} = 24$ms) to 0.5 (corresponding to $T_{idle} = T_{busy} = 4$ms) for three different packet size namely 40, 70, and 100 bytes.

As expected, increasing the duty cycle (i.e. decreasing the T_{idle}) increases the collision probability and therefore the packet reception rate decreases. The relation between PRR and the duty cycle of the interferer appears to be strongly linear and the PRR tends to 1 for duty cycle values close to zero (i.e. no collision for $T_{idle} \rightarrow \infty$). This, combined with the great reproducibility shown by the small error bars, allows us to state that the interference generator easily allows to induce any percentage of packets loss on a communication channel under study.

It is also interesting to note how different packet sizes correspond to different PRR values when leaving the interferer unchanged. In fact, increasing the packet size also increases the collision probability because of a higher time needed to transmit the packet which, consequently, reduces the PRR. The relation between packet length and communication performance under interference has been deeply investigated in [16]. It is also interesting to notice that no difference was found in the results between the case of the interferer configured with a modulated carrier and that with an unmodulated carrier. Hence, in both cases the interfering signal showed the same capability of disrupting packets intelligibility.

The comparison between the reliability values achieved through modeling equations and the packet reception rates measured experimentally is reported in Figure 5, which shows a good accordance between model (dashed lines) and real experiments (dotted lines) thus confirming a solid theoretical foundation to obtain predictable and reproducible experimental conditions.

Figure 6 compares the PRRs obtained in the previous experiment with those measured under two more realistic

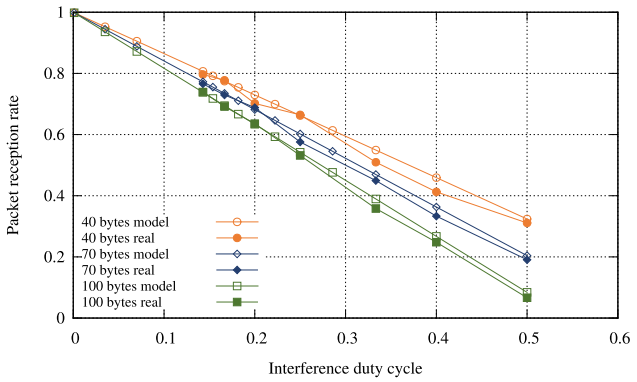

FIGURE 5. Comparison of experimental results with models.

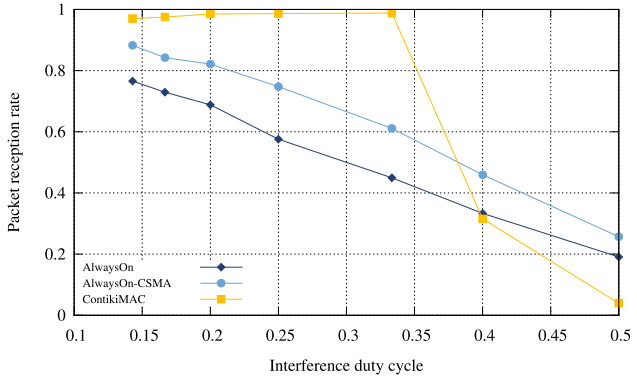

FIGURE 6. Packet reception rate when varying the interference duty cycle for different communication protocols.

nodes configurations, for varying values of the interferer duty cycle. In particular, the solid blue line plots the previous results for a packet size of 70 bytes and represents the base line while the dashed light blue line plots the PRR obtained when a CSMA/CA mechanism was enabled on the communication link. As expected, the collision avoidance strategy results in a more reliable communication for each value of the interferer duty cycle. Finally, when a low-power protocol such as ContikiMAC (yellow dotted line) is exploited for access to the medium, its performance seems to be particularly robust w.r.t. interference until the duty cycle reaches appreciably high values, where the PRR rapidly falls close to zero. This particular behavior is presumably due to different strategies implemented in ContikiMAC, namely the repeated CSMA/CA, and the multiple transmission of the same data packet. These ad-hoc strategies increase the channel reliability mitigating the effects of interference, at the expense of a reduced communication rate and of an increase in the energy expenditure of the sender node. Interestingly, when the duty cycle of the interferer continues to grow (beyond the values where its ad-hoc strategies are non longer effective), the ContikiMAC performance gets worse with respect to the null protocol because of bigger packet headers used by ContikiMAC.

Even when using the CSMA/CA strategy or the full ContikiMAC protocol, no appreciable changes in the results were measured in the modulated and unmodulated configurations of the interferer. Possible differences between the adoption of a randomly modulated or of an unmodulated signal in the busy state could arise not in terms of PRR but in terms of possible conflicts with energy optimisation strategies. For example, ContikiMAC implements a power saving technique called *fast sleep optimisation* which is aimed at helping the receiver node to determine if a busy channel condition was caused by noise rather than an incoming packet, so as to decide whether to turn off the receiver immediately or to wait for an incoming *Start Frame Delimiter* (SFD). This method, could be probably misled by a randomly modulated signal but not by an unmodulated one thus resulting, in the first case, in an increase of the energy consumption without directly affecting reliability.

B. BACKGROUND NOISE

In order to characterize the effects of the generated background noise over the wireless channel we conceived a set of experiments in which we measured the PRR and the RSSI during a broadcast communication while changing the transmission power of the interferer. The sensor node in charge of transmitting the broadcast packets was always configured with a transmission power of 0dBm while the interferer at each run increased its transmission power from $-13dBm$ up to $+7dBm$.

Notice that, the CC2538 radio transceiver uses a correlator to detect the start frame delimiter (SFD) and the corresponding correlation threshold value can be changed during initialization process by means of the appropriate register (namely `MDMCTRL1`). In other words, the correlation threshold determines how closely the received SFD must match an ideal SFD, so that adjusting it, the receiver sensitivity can be decreased or increased. Obviously, increasing the threshold effectively reduces the receiver sensitivity which can miss many actual SFDs. On the contrary, lowering the threshold can results in increased sensitivity mostly during interference conditions. Despite this, a too low value setting can lead to an opposite result (i.e. to a reduced sensitivity) because false frames might overlap with the SFDs of actual frames so as to prevent its detection. To better characterise the effectiveness of the proposed interferer we repeated the same experiment three times with three different value of the correlation threshold of the receiver node.

Figure 7 shows both the PRR and the RSSI measured during the experiments. The blue dashed curves refer to the left y axis and represent the PRR measured with increasing correlation threshold (TH1<TH2<TH3) while the solid light blue line shows the measured RSSI values and refers to the right y axis. As expected, increasing the interference power increases the RSSI while the corresponding PRR values show a strongly non-linear behavior. In fact, the PRR seems not to be affected by the increasing of the interference power until it rapidly collapses reaching soon the value of zero, which corresponds to the inhibition of each communication. Moreover, changing the correlation threshold shifts the PRR

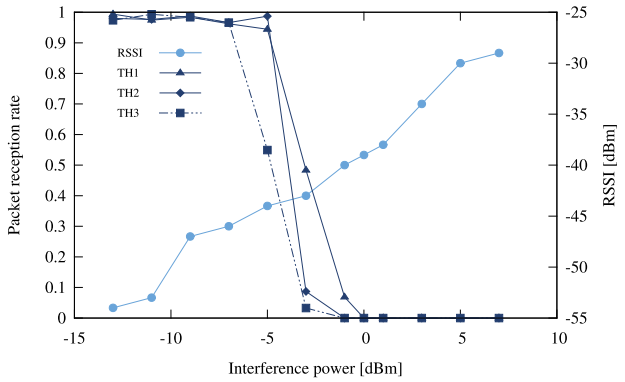

FIGURE 7. Packet reception rate and RSSI when vary the transmission power of the interferer.

fall in correspondence of higher interference power, for a small threshold, or in correspondence of lower interference power for a high threshold. Also in this case, the modulated and unmodulated configurations did not lead to appreciable differences in the measured values.

Thanks to these results it is possible to state that, despite the good capability of reaching desired tuning levels by adjusting the knob of the interference signal power (as demonstrated by the measured RSSI curve) the possibility of using this system for studying communication in WSNs subject to interference is hampered by the abrupt change of the PRR which shows a sort of binary regime.

C. DISCUSSION

The analysis of the experimental results allows to highlight some main findings. Regarding packet storm emulation, the reliability of the communication link linearly decreases with the duty cycle of the interference source; inverse proportionality w.r.t. packet size is also confirmed, as well as a higher robustness of communication protocols that adaptively defer transmissions (e.g. CSMA protocols), or that adopt more sophisticated strategies (e.g. ContikiMAC). For what concerns background noise emulation, the implemented system shows the capability to support fine tuning of RSSI levels as a function of the transmission power of the interference source. However, we also found that the corresponding packet reception rates present a transition phase with values rapidly falling from 1 to 0, which denotes the criticality of adopting background noise emulation when the effects of interference on communication link are assessed according to packet-related reliability metrics.

VI. CONCLUSIONS

Interference is widely recognized to potentially cause disruption in the operativeness of a WSN. In order to test the actual capability of a novel research methodology to cope with interference, methods that could guarantee repeatable, controllable and simple generation of ad hoc interference noise patterns are needed. In this work we have presented a thorough assessment of recent techniques for emulating interference sources by means of IEEE 802.15.4 radio transmitters mounted on board of 32bit-chip, low cost, tiny sensor nodes. This type of emulated signals belongs to the class of the so called in-band interference.

The experimental investigation has been conducted to measure the effects of in-band noise on the communication performance of a communication link. Indeed, we measured the performance of a system composed of two communicating sensor nodes subject to interference generated from a third one in terms of the packet loss rate, and under different possible configuration parameters. Two basic emulation typologies have been examined: *i)* the packet storm emulation and *ii)* the background noise emulation. Results shed light on some distinguishing features of the studied method which could help system designers to conceive experimental testing protocols. In particular: *i)* we measured the effect of the duty cycle of the interference source on the packet loss rate under different communication protocols; *ii)* we confirmed that in the background noise emulation mode the RSSI has a smooth dependence from the transmission power of the interference signal, while we showed that is prone to a sharp regime change in terms of packet loss rate (always as a function of the transmission power of the interference signal). This points out the potential ineffectiveness of the method when used to evaluate reliability metrics. To conclude, a few pointers to future work can be given. The availability of an affordable, reliable platform for precise generation of interference sources opens the way for experimental assessment of interference-robust communication protocols on large scale deployments. Another research direction that could be interesting to pursue is the implementation of probabilistic interference patterns that could model dynamic scenarios with varying levels of interference emulating, for instance, traffic bursts with statistically characterized arrival times [30].

REFERENCES

[1] X. Ma, P. Zhang, X. Li, W. Tang, J. Wei, and O. Theel, "DeCoT: A dependable concurrent transmission-based protocol for wireless sensor networks," *IEEE Access*, vol. 6, pp. 73130–73146, 2018.

[2] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli, "Low-power wireless for the Internet of Things: Standards and applications," *IEEE Access*, vol. 6, pp. 67893–67926, 2018.

[3] A. Bogliolo, E. Lattanzi, and V. Freschi, "Idleness as a resource in energy-neutral WSNs," in *Proc. 1st Int. Workshop Energy Neutral Sens. Syst.*, New York, NY, USA, Nov. 2013, p. 12.

[4] N. Baccour, A. Koubâa, L. Mottola, and M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Trans. Sensor Netw.*, vol. 8, no. 4, pp. 34:1–34:33, Sep. 2012. doi: 10.1145/2240116.2240123.

[5] C. A. Boano, Z. He, Y. Li, T. Voigt, M. Zúñniga, and A. Willig, "Controllable radio interference for experimental and testing purposes in wireless sensor networks," in *Proc. IEEE 34th Conf. Local Comput. Netw.*, Oct. 2009, pp. 865–872.

[6] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñniga, "JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2011, pp. 175–186.

[7] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher, "RID: Radio interference detection in wireless sensor networks," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 2, Mar. 2005, pp. 891–901.

[8] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, Oct. 2006, pp. 237–250.

[9] E. Toscano and L. L. Bello, "Cross-channel interference in IEEE 802.15.4 networks," in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, May 2008, pp. 139–148.
[10] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens, "Dependable interference-aware time-slotted channel hopping for wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 14, no. 1, p. 3, Mar. 2018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3176363.3158231
[11] A. Saifullah, S. Sankar, J. Liu, C. Lu, R. Chandra, and B. Priyantha, "CapNet: Exploiting wireless sensor networks for data center power capping," *ACM Trans. Sensor Netw.*, vol. 15, no. 1, p. 6, Dec. 2018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3300201.3278624
[12] Ettus Research LLC. *Universal Software Radio Peripheral (USRP)*. Accessed: Jul. 17, 2018. [Online]. Available: http://www.ettus.com
[13] Q. Lin and M. A. Weitnauer, "Practical residual interference after cancellation for constant envelope modulation with data-aided synchronization," *IEEE Access*, vol. 6, pp. 69230–69241, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8529182/
[14] J. Yao, W. Lou, C. Yang, and K. Wu, "Efficient interference-aware power control for wireless networks," *Comput. Netw.*, vol. 136, pp. 68–79, May 2018. doi: 10.1016/j.comnet.2018.02.017.
[15] Texas Instruments. *CC2538 Powerful Wireless Microcontroller System-On-Chipfor 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee Applications*. Accessed: Jul. 17, 2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2538.pdf
[16] V. Freschi and E. Lattanzi, "A study on the impact of packet length on communication in low power wireless sensor networks under interference," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3820–3830, Apr. 2019.
[17] J.-S. Han and Y.-H. Lee, "Interference-robust transmission in wireless sensor networks," *Sensors*, vol. 16, no. 11, p. 1910, 2016.
[18] Texas Instruments. *SmartRF06 Evaluation Board*. Accessed: Jul. 17, 2018. [Online]. Available: http://www.ti.com/tool/SMARTRF06EBK
[19] Texas Instruments. *SmartRF Studio*. Accessed: Jul. 17, 2018. [Online]. Available: http://www.ti.com/tool/smartrftm-studio
[20] U. Raza, A. Bogliolo, V. Freschi, E. Lattanzi, and A. L. Murphy, "A two-prong approach to energy-efficient WSNs: Wake-up receivers plus dedicated, model-based sensing," *Ad Hoc Netw.*, vol. 45, pp. 1–12, Jul. 2016.
[21] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Tampa, FL, USA, Nov. 2004, pp. 455–462. doi: 10.1109/LCN.2004.38.
[22] N. Brouwers, K. Langendoen, and P. Corke, "Darjeeling, a feature-rich vm for the resource poor," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst.*, New York, NY, USA, Nov. 2009, pp. 169–182. doi: 10.1145/1644038.1644056.
[23] E. Lattanzi, V. Freschi, and A. Bogliolo, "Supporting preemptive multitasking in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 2, 2014, Art. no. 814510.
[24] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, New York, NY, USA, Oct. 2006, pp. 307–320. doi: 10.1145/1182807.1182838.
[25] A. Dunkels, "The ContikiMAC radio duty cycling protocol," Swedish ICT Res. Softw. Inst., Kista, Sweden, Tech. Rep. T2011:13, 2011, pp. 1–11. [Online]. Available: http://dunkels.com/adam/dunkels11contikimac.pdf
[26] R. Musaloiu-E and C.-J. M. Liang, and A. Terzis, "Koala: Ultra-low power data retrieval in wireless sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2008, pp. 421–432. [Online]. Available: http://ieeexplore.ieee.org/document/4505492/
[27] Rohde & Schwarz. *R&S NGMO2–Brochure e Schede Tecniche*. Accessed: Jul. 17, 2018. [Online]. Available: https://www.rohde-schwarz.com/it/brochure-scheda-tecnica/ngmo2/
[28] National Instruments. *Device Specifications NI 6251*. Accessed: Jul. 17, 2018. [Online]. Available: http://www.ni.com/pdf/manuals/375213c.pdf
[29] National Instruments. *Installation Guide BNC-2120*. Accessed: Jul. 17, 2018. [Online]. Available: http://www.ni.com/pdf/manuals/372123d.pdf
[30] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting WiFi white space for ZigBee performance assurance," in *Proc. 18th IEEE Int. Conf. Netw. Protocols*, Oct. 2010, pp. 305–314.

EMANUELE LATTANZI received the Laurea degree *(summa cum laude)* and the Ph.D. degree from the University of Urbino, Italy, in 2001 and 2005, respectively. Since 2001, he has been with the Information Science and Technology Institute, University of Urbino, Italy, where he has been an Assistant Professor of information processing systems with the Department of Pure and Applied Sciences (DiSPeA), since 2008. In 2003, he was with the Department of Computer Science and Engineering, Pennsylvania State University, as a Visiting Scholar with Prof. V. Narayanan. His research interests include wireless sensor networks, wireless embedded systems, energy-aware routing algorithms, dynamic power management, multimedia applications, and simulation.

VALERIO FRESCHI received the Laurea degree in electronic engineering from the University of Ancona, Italy, in 1999, and the Ph.D. degree in computer science engineering from the University of Ferrara, Italy, in 2006. He is currently a Research Fellow of computer engineering with the Department of Pure and Applied Sciences (DiSPeA), University of Urbino, Italy. His research interests include wireless sensor networks, networked embedded systems, graph algorithms, bioinformatics, and optimization.