Controlled Interference Generation for Wireless Coexistence Research

Anwar Hithnawi, Vaibhav Kulkarni, Su Li, Hossein Shafagh
Department of Computer Science
ETH Zurich, Switzerland
{hithnawi, kvaibhav, lisu, shafagh}@inf.ethz.ch

ABSTRACT

In recent years, we have witnessed a proliferation of wireless technologies and devices operating in the unlicensed bands. The resulting escalation of wireless demand has put enormous pressure on available spectrum. This raises a unique set of communication challenges, notably co-existence, Cross Technology Interference (CTI), and fairness amidst high uncertainty and scarcity of interference-free channels. Consequently, there is a strong need for understanding and debugging the performance of existing wireless protocols and systems under various patterns of interference. Therefore, we need to augment testbeds with tools that can enable repeatable generation of realistic interference patterns. This would primarily facilitate wireless coexistence research experimentation. The heterogeneity of the existing wireless devices and protocols operating in the unlicensed bands makes interference hard to model. Meanwhile, researchers working on wireless coexistence generally use interference generated from various radio appliances. The lack of a systematic way of controlling these appliances makes it inconvenient to run experiments, particularly in remote testbeds. In this paper, we present a Controlled Interference Generator (CIG) framework for wireless networks. In the design of CIG, we consider a unified approach that incorporates a careful selection of interferer technologies (implemented in software), to expose networks to realistic interference patterns. We validate the resemblance of interference generated by CIG and interference from represented RF devices, by showing the accuracy in temporal and spectral domains.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids.

Keywords

Cross Technology Interference, GNU Radio, Software Defined Radios, Wireless Coexistence Experimentation

SRIF'15, September 7, 2015, Paris, France.
© 2015 ACM. ISBN 978-1-4503-3532-4/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2801676.2801682.

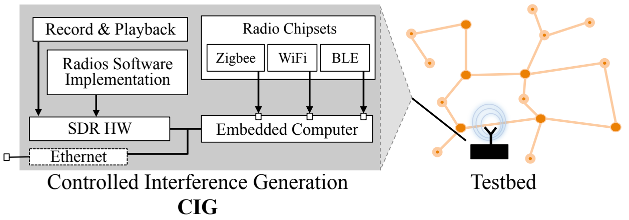


Figure 1: Schematic of our Controlled Interference Generation (CIG) framework, facilitating advanced wireless coexistence experimentation.

1. INTRODUCTION

The ubiquitous and tetherless access to information that the wireless medium is enabling and recent advances in wireless communication have led to a rapid surge in wireless data traffic congesting the unlicensed bands. This traffic is generated from heterogeneous radios that follow different protocols and communication primitives. A few examples include WiFi (IEEE 802.11), Bluetooth, IEEE 802.15.4, 2.4 GHz cordless phones, surveillance cameras, game controllers, and 2.4 GHz RFID. With the proliferation of wirelessly connected devices, coupled with rapid increase in newly emerged radios [1], it is crucial to understand how CTI can impact the performance of wireless networks and emerging pervasive RF-based services, such as indoor localization [16, 18, 20] and activity recognition systems [5, 6]. Independent academic and industrial studies [2, 3, 9, 11, 24] show that wireless networks and RF-based systems experience non-negligible performance degradation due to CTI. The impact of CTI on low-power wireless networks is even more severe due to their low transmission power. These networks suffer to coexist and compete for the shared channel access.

To improve the interference robustness of wireless systems, it is beneficial to gain a detailed understanding of how heterogeneous wireless systems and networks coexist and operate in the crowded unlicensed spectrum. Therefore, it is essential to augment testing environments with a *repeatable, controllable*, and *realistic* interference generation. Researchers working on wireless coexistence, either use modeling and simulation [22, 31], which are typically abstract and less accurate, or use interference generated from actual wireless devices [8, 10, 24]. While the latter approach is more realistic, it is costly, labor intensive, and impractical as some of these devices can not be controlled in a systematic way (e.g., microwave oven, analog phone, etc.), especially when experiments are run in remote testbeds. Jamlab [14], a recent approach, makes use of commodity hardware by utilizing a subset of the nodes in the testbed to generate controllable interference patterns. However, such systems have shortcomings in accuracy and the range of interference types they can support. Due to hardware limitations, such approaches are restricted to the fixed modulation schemes supported by the nodes used in the testbed and limited to the rate at which frequency hopping can be performed.

In this paper, we present **CIG**, a SDR design for controlled interference generation, which can facilitate augmenting current testbeds with repeatable and realistic interference pattern generation (see Figure 1). CIG provides three modules for interference generation: *(i) Record and Playback*; this module features high precision record and playback. It can be used to record and playback various interferer patterns, but is particularly interesting for devices that are not feasible to be implemented in SDR, such as microwave ovens, and proprietary radios where we lack the know-how on their physical layer implementation. *(ii) Radio Software Implementation*; this module allows generation of interference from radios (i.e., the physical layer) implemented in software. For this, we implement or port radios of a set of prevalent interferers on a Universal Software Radio Peripheral (USRP). This set includes commercially available analog cordless phones, digital FHSS phones, security cameras, baby monitors, WiFi, and ZigBee devices. *(iii) Commercial Radio Chipsets*; this module allows the generation of interference patterns from a subset of commercial radio chipsets that are interfaced with an embedded computer within CIG. This further allows us to cover commercial software and hardware artifacts of different radio chipsets.

CIG is not bound to the set of interferer technologies presented in this paper and each of its modules can be extended to include new technologies. We provide a unified, simple to use interface for controlling CIG through command-line host software. We perform an initial validation of the generated interference patterns by correlating the generated and real interference in time and frequency domains. Furthermore, we analyze the impact of generated interference on low-power networks to ensure accuracy and similarity to the interference patterns from original RF interferers. Moreover, we provide insights on limitations and challenges of bringing some commercial radios to SDR.

2. DESIGN OVERVIEW OF CIG

We now present a high-level design overview of CIG, as illustrated in Figure 2. CIG provides three modules to generate controllable interference. In this prototype of CIG, we focus on incorporating a set of interferer technologies that are prevalent in the unlicensed bands. Our considered set of interferers covers low/high power, narrow/wide band, analog/digital, and channel hopping/fixed frequency interferers. This set represents common underlying properties adopted by most radio technologies.

Record and Playback. This module of CIG is realized on SDR and allows recording temporal and spectral patterns of a particular interference and playing back these patterns as energy pulses emitted in the spectrum. For a large body of interference mitigation research, particularly solutions residing in MAC and upper layers (e.g., clear channel assessment, interference avoidance, channel sampling for free channel discovery, and channel occupancy patterns for opportunistic MAC scheduling) it is sufficient to focus on temporal and spectral characteristics of interferers. The modulated signal type thereby is of less relevance. Moreover, interferers that are not inherently RF radios, such as microwave oven or closed radios, which cannot be implemented on SDR, are appropriate candidates to be represented through the playback module of CIG.

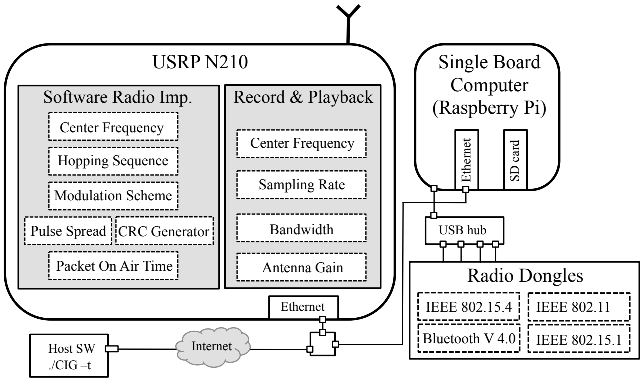


Figure 2: Architecture of CIG. The *Software Radio Implementations* and *Record and Playback* modules reside on USRP N210. The single board computer enables generation of interference from off-the-shelf radio dongles.

Software Radio Implementations. This module allows interference generation of a set of prevalent interferers. We enable this by implementing the wireless stack of these interferes in SDR, while aiming to achieve an authentic physical layer behavior. This module can be used while developing interference mitigation schemes where the type of modulated interfering signal is relevant. This is particularly relevant with physical layer solutions, such as, interference source classification [25], interference suppression, and cancelation [24]. Moreover, it allows verifying whether emerging radios [32] and wireless systems can cause harm for competing technologies and quantify the impact.

Commercial Radio Chipsets. Reaching hardware-like efficiency and predictability with software implementation of wireless stacks on SDRs is challenging and not always feasible. With this module, we have the possibility of generating interference from standard off-the-shelf radio chipsets. Thus, it allows covering the impact of commercial software and hardware artifacts of different radio chipsets and overcoming limitations of SDRs, namely: *(i)* Due to strict timing requirements, carrier sensing is hard to implement in software (e.g., 802.11 backoff). *(ii)* Due to strict frequency tuning capabilities, it is hard to achieve high frequency hopping rate in software (e.g., Bluetooth exhibits a hopping rate of 1600 hops/s).

3. REALIZATION

In this section, we elaborate on CIG's hardware and software architecture. We first give a brief overview of our platform and then discuss implementation aspects of modules.

3.1 Platform

The hardware platform of CIG consists of two main components (see Figure 2). The main component is a SDR where the *Record and Playback* and *Software Radio Implementations* are realized. The second component is a low-power computer that controls the *Commercial Radio Chipsets*.

We provide a unified interface in the form of extendable scripts that interact with the corresponding CIG component to generate interference. The interface is typically connected via Internet to the main CIG platform, located in a testbed.

SDR Component. For the SDR hardware, we rely on the Ettus USRP N210 [29], which is equipped with 100 M samples/s 14-bit ADCs and 400 M samples/s 16-bit DACs. It is connected to a host computer via a Gigabit ethernet port and can stream up to 25 M samples/s to/from host applications. For the RF front-end, we use the SBX radio daughterboard [26]. The SBX board incorporates a wide band transceiver that operates from 400 MHz to 4400 MHz. It provides up to 40 MHz of instantaneous usable bandwidth and up to 100 mW of transmission power.

For development, we rely on GNU Radio [17], an open source software toolkit for building software radios. GNU Radio provides libraries for signal processing blocks. In order to build a typical wireless radio stack, flow graphs, composed of a sequence of *Digital Signal Processing* (DSP) blocks, are created (see Figure 3). Moreover, a state machine selects the corresponding flow graph to process incoming samples. These DSP blocks are created in C++ and connected in a python wrapper to build the flow graphs. For example, the receiver of a DSSS analog phone has blocks for clock synchronization, channel equalization, Costas loop for phase and frequency correction, BPSK demodulator, symbol to constellation mapper and direct-sequence despreader. Different blocks are integrated into separate flow graphs, each addressing different communication tasks, such as ACK packets, and inbound and outbound communication. In the last step, the flow graphs are assembled into a DSSS cordless phone receiver state machine.

Embedded Computer Board. We use a Raspberry Pi as a single-board embedded computer which hosts a quad-core ARM Cortex-A7 controller [23]. It serves as a low cost and small form factor linux platform to interface off-the-shelf radio chipsets, as illustrated in Figure 2.

3.2 Record and Playback

Now we describe how to conduct RF record and playback using the USRP.

Interferers. The *Record and Playback* module is not bound to any specific interferer. This module can be used to record and playback RF radio technologies or playback (third party) recorded files or synthesized RF signals. We record RF signals of three interfering technologies operating in the unlicensed bands, namely: *(i)* Microwave oven,*(ii)* Analog DSSS cordless phone, and *(iii)* Wireless camera. We refer to Table 1 for technical details about the interferer devices used in this project. We select these particular technologies, representing three typical CTI behaviors, namely: frequency sweeping, frequency static, and high rate frequency hopping, respectively, to analyze the system's record and playback capability.

Record. We record 50 million samples by configuring the USRP to tune to the respective device's operational bandwidth and center frequency (f_c), as listed in Table 1. We perform the recording in an office environment. However, to maximize the correlation between the recorded and the actual signal, the recording can be performed in an anechoic chamber, which ousts the impact of nearby interfering signals on the recorded signal.

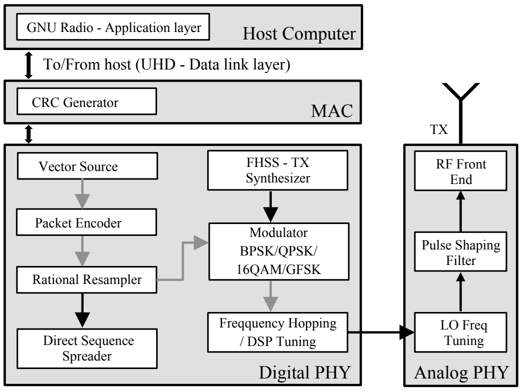


Figure 3: Simplified USRP block diagram to signal flow graph mapping. As an example, USRP implementation of the wireless camera is indicated by DSP blocks connected with gray arrows.

While the center frequency and the bandwidth need to be adjusted according to the wireless radio specifications of the interferer, the receive gain parameter needs to be adjusted according to the peak power of received signal and the SDR hardware specifications (i.e., the supported ADC range). The receive gain influences the accuracy of recorded signal, thus need to be adjusted to attain a unit amplitude of the recorded baseband signal, in order to use the full range of the 14-bits ADC without clipping. This does not necessarily correspond to the highest gain. For instance, recording a high-power microwave oven at 1 m distance, with the maximum gain of SBX (31.5 dB), results into signal clipping. Hence, it is necessary to select the receive gain in such a way that the clipping is avoided. For example for microwave oven, the receive gain of 25 dB avoids clipping at 1 m distance.

Playback. The recorded signals are stored as 16 bit I/Q data samples. During playback, the recorded raw baseband data is sent to the USRP, which converts it to analog signal. The analog signal is then transmitted by the USRP by up-converting it to the RF signal. We configure the USRP's data rate (i.e., the rate of reading the recorded file) to match the recording sampling rate. The f_c is set according to the device specifications. The SBX daughter board has a nonlinear gain response when operating in a wide bandwidth [27]. Therefore, it is challenging to regenerate the wide-band recorded signal at the accurate power level, as the down-converted baseband signal does not match the actual transmit power specifications of the device. Hence, during playback, we set the transmit gain value to match the average power level and the peak power to the specified signal power (according to device specifications).

The accuracy of the playback signal is dependent upon hardware limitations of USRP, particularly the sampling rate, maximum transmit power, frequency tuning and settling time, and latency in the hopping rate imposed by the OS scheduling and Ethernet transmission time. We observe that the *Record and Playback* module is suitable for narrow band interferers occupying static frequency channels, e.g., the DSSS cordless phone [7], provided that adequate device specifications are available to set the recording parameters. It is also suitable for frequency sweeping microwave ovens where the sweep to the next frequency channel

RF Technology	Vendor & Product Name	TX Power (dBm)	Channel Width (MHz)	Modulation Scheme	Spectrum Range (GHz)
Analog Phone	Vtech GZ2456	n/a	0.1 (Static)	DSSS and BPSK	2.41 - 2.42
Analog Phone	Uniden TRU 4465-2	n/a	0.08 (Static)	DSSS and GFSK	2.40 - 2.48
FHSS Cordless Phone	Uniden DCT6485-3HS	21	0.8 (FH)	GFSK and FHSS	2.41 - 2.47
Wireless Camera	Philips SCD 603	20	1.125 (FH)	BPSK	2.42 - 2.46
Wireless Camera	Genica C-501	20	0.1 (Static)	GFSK	2.41 - 2.47
IEEE 802.15.4	XBee XBP24-AWI-001	4	2 (Static)	DSSS and O-QPSK	2.40 - 2.48
Bluetooth (Class 2)	Bluetooth V2.0 EDR	4	1 (FH)	GFSK	2.40 - 2.48
BLE (Bluetooth V.4.0)	BLED112	4	2 (FH)	GFSK	2.40 - 2.48
Microwave Oven	Clatronic MWG 758	60	-	-	2.44 - 2.48
IEEE 802.11	RTL8192cu Chipset	17	20 (Static)	DSSS, DBPSK	2.40 - 2.48

Table 1: Characteristics of considered RF technologies supported by CIG.

typically occurs after 10-15 ms which provides sufficient time to the USRP for retuning and settling to the next frequency. We observe that the USRP is accurately able to capture the on-off patterns of the microwave oven, over 40 MHz of bandwidth. However, for frequency hopping interferers, such as wireless cameras where the typical hopping rate is 400-600 hops/s, the frequency synthesizer is not able to capture all the packets, switch and settle to the next hop frequency in a bounded time to accurately represent the device specific frequency hopping nature.

3.3 Software Radio Implementations

We implement the physical layer (PHY) of five commercially available wireless interfaces, operating in the unlicensed bands. We use the GNU Radio [17] framework to build the signal processing blocks and construct flow graphs of the considered radios. In case of proprietary technologies, we implement the physical layer according to the description in the devices manuals and the spectral analysis. Figure 3 shows the implemented flow graph of the wireless camera, as an example. Additionally, we implement a CRC generator in the sender device and a CRC checker in the receiver device, to create statistics about the performance of the transmission. This enables researchers to quantify the harm of their solutions on other competing devices, such as wireless cameras where so far it is not trivial to quantify this impact. In the following, we elaborate on the implemented PHYs:

Analog Cordless Phone. Our analog cordless handset [15] operates in a narrow frequency band [2410.2 - 2418.9] MHz. The user can configure the device to operate on one of 30 supported channels, each 100 kHz wide. The phone uses DSSS to spread the BPSK modulated data. We use a vector source to generate bit streams followed by the spread spectrum block and connect the output to a BPSK modulator. We set the center frequency of the USRP sink block to match the f_c of the first supported channel (2.417 GHz). The transmitting channel is configurable through the host software.

DSSS Cordless Phone. The phone base and the handset [7] communicate using digital spread spectrum and operate in the frequency band [2.407 - 2.478] GHz. The phone supports 28 possible channels, each 3 MHz wide, and shifts the operational channel automatically upon sensing interference. In our implementation, we provide the channel selection option to the user. The phone uses a data rate of 1.366 Mbit/s [24], employs digital spread spectrum, and transmits the data over GFSK modulation. We use the rational re-sampler block to achieve the specified data rate. The interpolation and decimation values can be derived from Equation 1 where the desired bit rate depends on the DAC sampling rate and the number of Samples per Symbol (SPS). We further connect this block to the DSSS block and GFSK modulator.

$$Bit\ Rate = DAC\ Rate/(Interpolation \times SPS) \qquad (1)$$

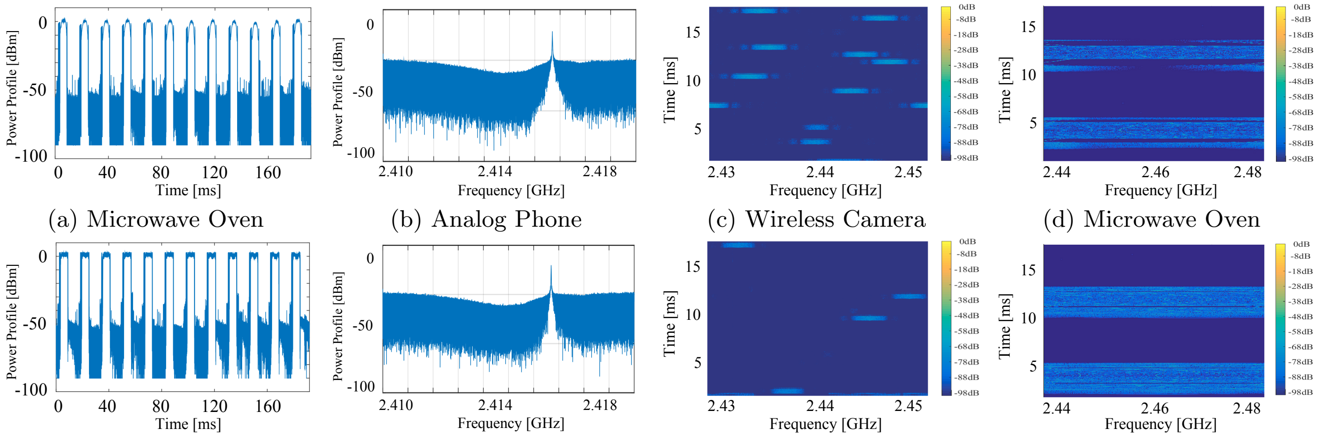
Wireless Camera. We consider integrating two wireless cameras [4, 21]. The first wireless baby monitor [21] communicates with the video receiver using frequency hopping over 61 channels, each of which has a bandwidth of 1.125 MHz and uses BPSK modulation scheme. The second wireless monitoring camera [4] supports 4 different channels (2.414 GHz, 2.432 GHz, 2.450 GHz, and 2.468 GHz) and occupies a wide bandwidth of 16 MHz. We perform spectral analysis of these technologies to examine the on-air packet time, hopping sequence, and hopping rate. For the Philips baby monitor, we observe the average packet on-air time to be 2.2 ms with a hopping rate of 450 hops/s.

We use respective blocks to generate packets and modulate them as specified in the device specifications. We connect the modulated output to the frequency hopping block. The USRP N210 has two stages of frequency tuning: *(i)* RF front-end which translates between the RF and the intermediate frequency (IF), and tunes the frequency as close as possible to f_c. *(ii)* DSP, which translates from the IF to the baseband, accounts for the error in frequency tuning, and digitally sets the necessary offset to tune to the desired f_c. In order to achieve faster hopping rates in the order of 2 ms tuning time, we fix the RF front-end frequency at the center of the band and hop via shifting in the FPGA only by using timed transactions and tune request objects [30]. We generate the signal at baseband and use the FPGA to convert the signal digitally to the correct frequency. We also schedule the frequency changes and streaming commands a priori to hop faster and deterministically, using timed transactions. We set the channel changes to cover all the channels specified within the operational bandwidth. The time is set to achieve the maximum number of hops possible through our implementation which is 280 hops/s.

FHSS Cordless Phone. The phone base and handset [28] communicate using FHSS, hopping over 90 channels in the range [2.4075 - 2.472] GHz, with a channel width of 800 kHz and GFSK modulation. The discussion we provided on the wireless camera implementation applies here too, given that both technologies employ the same underlying signal spreading scheme, i.e., frequency hopping, only with slight changes in channel bandwidth and hopping rate.

3.4 Commercial Radio Chipsets

To generate traffic of prevalent communication standards, we use radio chipsets of various technologies, such as, IEEE 802.11 (b/g/n) [19], Bluetooth class 2 [13], Bluetooth Low Energy [12], and ZigBee [33]. The transmission power,



(a) Microwave Oven (b) Analog Phone (c) Wireless Camera (d) Microwave Oven

(e) CIG Microwave Oven (f) CIG Analog Phone (g) CIG Wireless Camera (h) CIG Microwave Oven

Figure 4: Comparison of interference patterns of actual interferers in the first row and CIG in the second row. (a) and (e) depict time profiles of the microwave oven. (b) and (f) depict periodograms of the analog phone. (c) and (f) depict spectrograms of the wireless camera. (d) and (h) depict spectrograms of the microwave oven.

channel number, and traffic parameters can be configured by the user via the host software to emulate various application traffic patterns.

4. VALIDATION

We perform an initial validation of CIG in the time and frequency domains. We also quantify the impact of the interference generated by CIG as opposed to real interferers, by subjecting an 802.15.4 communication link to both generated and real interference. The experimental setup consists of 2 prototypes of CIG, 2 low-power sensors nodes (TelosB) forming the communication link, and the discussed interferer technologies.

4.1 Temporal Accuracy

We evaluate the accuracy of interference generated by CIG compared to the real interference in the time domain. For this, we record the interference signal from CIG and the corresponding interferer device. Afterward, we compare the pulse duration and number of pulses in a given time period for each technology. For instance as depicted in 4(a) and 4(e) for microwave oven, we observe equal number of pulses and similar timing behavior. In order to quantify the accuracy, we cross-correlate the playback signal with respect to the recorded signal. We convert the signal series to binary values, where 0 stands for clear channel and 1 for a busy channel, given a threshold of -45 dBm (typical CCA threshold for 802.15.4). The cross-correlation coefficient c can thus be represented by:

$$c = \frac{1}{N}\sum_{i=1}^{N} x(i) \odot y(i) \qquad (2)$$

Where N is the number of samples, $x(i)$ and $y(i)$ are original (recorded) and played back signals, respectively, with $i = [1, 10^6]$. For microwave oven, where the signal exhibits an on and off pattern, the average cross-correlation coefficient c over the length of the samples is 0.926 with a standard deviation of 0.0764. This high accuracy is due to the good performance of SDR in playing back the recorded samples without a noticeable jitter. In case of analog DSSS phone, we observed a high cross-correlation value of 0.998. The wireless camera uses frequency hopping, hence to validate its temporal behavior, we compare the on-air packet time and the number of packets generated in a given time frame. Figures 4(c) and 4(g) visualize the general trend. We observe an average cross-correlation coefficient of 0.930 for each packet. However, we reach only 62.2% of the required hopping rate which is due to hardware limitations of SDR, as discussed in Section 3.

4.2 Spectral Accuracy

In order to quantify the spectral accuracy of CIG, we consider aspects representing particular spectral behavior of the considered interferers. That is the static frequency behavior of analog phone where the signal peak lies at the center frequency of the selected channel, the frequency sweeping behavior of microwave oven where the sweeping occurs within the second half of the ISM band, and frequency hopping behavior of the wireless camera. We analyze the power spectral density and consider 95% occupied bandwidth for comparison. We compare the center frequency of the signal in case of the analog phone which lies at 2.417 GHz in both cases (see Figure 4(b) and 4(f)). The occupied bandwidth is 100 kHz for the actual phone and 107 kHz for CIG showing a reasonable accuracy for analog phone. In case of microwave oven, we validate the frequency sweeping behavior by comparing the spectrograms of the actual microwave and CIG, depicted in Figure 4(d) and 4(h). We observe a high energy present on the channel corresponding to microwave on cycles for both of the cases. The average bandwidth occupied by the on cycle amounts to approximately 284 kHz for the actual microwave and the played back signal by CIG. In case of wireless camera it is challenging to compare and validate the channel switching pattern used in frequency hopping due to absence of a particular sequence, hence we only compare the average bandwidth occupied by each packet which is 2.22 MHz for actual camera and 2.38 MHz for CIG.

4.3 Impact on Communication Link

In the following, we study the impact of interference on the performance of an 802.15.4 link subjected to interference generated by actual devices and as compared to CIG. For the communication link, we use a pair of TelosB nodes. We evaluate various setups, but highlight here the following one: The transmitter sends 1000 packets, each with a length of 50 bytes and CCA enabled at a transmit power of 0 dBm with an interval of 100 ms to a receiver placed 4 m away.

The transmitter logs CCA status before each transmission. The receiver logs statistics about received packets including RSSI, LQI reading, and the induced power level on the channel. We select the communication channel to overlap the one on which the interference sources are active, or the one within the frequency hopping sequence.

In our experiments, CIG exhibits in most cases similar impact on the communication link as the real devices. The packet reception rate (PRR) obtained for CIG's microwave oven, is 6.2% lower than the original oven. This is due to USRP's transmit power adjustment during signal playback which results in an increased noise level at the off periods of the microwave oven signal. This consequently leads to slightly higher packet losses for receivers at distances affected by the residual noise. Similarly, we observe a lower LQI (indicating bad link quality), and higher noise readings, which only vary within 2 dBm.

In case of the analog phone, the 802.15.4 transmitter keeps backing off thus communication was not possible. This is due to the phone continuously emitting energy in the medium, thus monopolizing it completely. For both CIG and the actual device, we measure similar noise level and LQI values. While disabling CCA (as explored by [9] to allow communication during persistent inference), CIG results into similar performance as the actual device. Hereby, the PRR remains almost the same, showing a reasonable accuracy for analog phone interferer.

In case of the wireless camera, the PPR is 13.3% higher in case of CIG generated interference. This is due to the hopping rate limitations and consequently lower packet transmission rate. The average LQI and noise values for both interferers are, however, in the same range. The high LQI with wireless camera, indicating good links, is due to the frequency hopping nature of the camera. Then, on each channel in the hopping sequence, only for a short duration energy is emitted. We measure similar average RSSI values (variance of ±2 dBm) during packet reception, in both cases.

5. CONCLUSION AND FUTURE WORK

Radio frequency interference has a significant impact on the performance of wireless networks and RF-based wireless systems. To allow testing wireless communication protocols and systems under various interference patterns, we need to augment testbeds and experimental environments with tools that are capable of generating realistic and repeatable interference patterns, and yet easy to access and use. In this paper, we introduce CIG, a controlled interference generator implemented using SDRs. CIG incorporates the implementation of a set of prevalent radio interferers in one device that can be installed in remote testbeds. CIG incorporates playback capabilities to regenerate recorded interference patterns, as well as software radio implementation of a set of prevalent interferers operating in the unlicensed band. CIG is easy to use, install, and configure. We validate the spectral and temporal accuracy of the interference generated by CIG. Currently, we are planing to augment a public testbed with CIG, in order to perform a thorough evaluation and validation of CIG under various scenarios.

6. REFERENCES

[1] FCC Lab: Report On Trends in Wireless Devices. www.fcc.gov/oet/info/documents/reports/wirelessdevices.doc.

[2] Estimating the Utilisation of Key License-Exempt Spectrum Bands, Final Report, Mass Consultants Ltd., Ofcom, 2009.

[3] Miercom: Cisco CleanAir Competitive Testing, Lab Test Report DR100409D, Miercom, 2010.

[4] 2.4 GHz 4-Channel Wireless Receiver and 4 Wireless Infrared Color Cameras, Genica. www.genica.com.

[5] Q. Pu, S. Gupta, S. Gollakota, S. Patel. Whole-home Gesture Recognition Using Wireless Signals. In *ACM MobiCom*, 2013.

[6] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, H. Liu. E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures. In *ACM MobiCom*, 2014.

[7] 2.4 GHz Cordless Telephone. http://cdn-media-att.vtp-media.com/ecp/documents/ product-Product/391/UserManual/2173/e2725b_manual_bkm.pdf.

[8] A. Hithnawi. Exploiting Physical Layer Information to Mitigate Cross-Technology Interference Effects on Low-Power Wireless Networks. In *ACM SenSys*, 2013.

[9] A. Hithnawi, H. Shafagh, S. Duquennoy. Understanding the Impact of Cross Technology Interference on IEEE 802.15.4. In *ACM WiNTECH*, 2014.

[10] A. Hithnawi, H. Shafagh, S. Duquennoy. TIIM: Technology-Independent Interference Mitigation for Low-power Wireless Networks. In *ACM/IEEE IPSN*, 2015.

[11] B. Wei, W. Hu, M. Yang, C. T. Chou. Radio-based Device-free Activity Recognition With Radio Frequency Interference. In *ACM/IEEE IPSN*, 2015.

[12] BLED112. https://www.bluegiga.com/en-US/products/bled112-bluetooth-smart-dongle/.

[13] Bluetooth Dongle. https://www.hama.com/items/00092498.

[14] C. A. Boano, T. Voigt, C. Noda, K. Romer, M. A. Zuniga. JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In *ACM IPSN*, 2011.

[15] Coexistence between ZigBee and Other 2.4 GHz Products. http://www.atmel.com/Images/Atmel-42190-Coexistence-between-ZigBee-and-Other-24GHz-Products_AP-NoteAT02845.pdf.

[16] F. Adib, Z. Kabelac, D. Katabi. Multi-Person Localization via RF Body Reflections. In *USENIX NSDI*, 2015.

[17] GNU Radio Companion. https://gnuradio.org/.

[18] H. Shafagh. A. Hithnawi. Poster: Come Closer - Proximity-based Authentication for the Internet of Things. In *MobiCom*, 2014.

[19] IEEE 802.11 Dongle. http://www.adafruit.com/product/1030.

[20] M. Youssef, M. Mah, A. Agrawala. Challenges: Device-free Passive Localization for Wireless Environments. In *ACM MobiCom*, 2007.

[21] Philips SCD 603 digital video baby monitor. http://www.usa.philips.com/c-p/SCD603_10/avent-digital-video-baby-monitor.

[22] R. Neumeier, G. Ostermayer. Analyzing Coexistence Issues in Wireless Radio Networks – The Simulation Environment. In *Modelling Symposium (EMS)*, 2013.

[23] Raspberry Pi Model B+. https://www.raspberrypi.org/.

[24] S. Gollakota, F. Adib, D. Katabi, S. Seshan. Clearing the RF Smog: Making 802.11n Robust to Cross-technology Interference. In *SIGCOMM*, 2011.

[25] S. Hong, S. Katti. DOF: A Local Wireless Information Plane. In *ACM SIGCOMM*, 2011.

[26] SBX 400-4400 MHz. www.ettus.com/product/details/SBX.

[27] SBX Nonlinearity. http://files.ettus.com/performance_data/sbx/SBX-without-UHD-corrections.pdf.

[28] Uniden DCT6485-3HS cordless handset system. http://www.uniden.com/pdf/DCT6485om.pdf.

[29] USRP N210. www.ettus.com/product/details/UN210-KIT.

[30] USRP Tuning Process. http://files.ettus.com/manual/page_general.html/general_tuning_process.

[31] V. Iyer, M. Woehrle, K. Langendoen. Chamaeleon: Exploiting Multiple Channels to Mitigate Interference. In *INSS*, 2010.

[32] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, J. R. Smith. Ambient Backscatter: Wireless Communication out of Thin Air. In *ACM SIGCOMM*, 2013.

[33] XBee Pro Module. http://www.adafruit.com/products/964.