iSCISM: Interference Sensing and Coexistence in the ISM Band

By Joe Baylon, Ethan Elenberg, Samantha Massengill
Department of Electrical Engineering
The Cooper Union for the Advancement of Science and Art

There is an ever-increasing need for effective interference-mitigation schemes in WiFi networks.

1. Introduction

A multitude of commercial devices transmit in the unlicensed 2.4 GHz ISM band, and these devices are likely to interfere with each other. These effects are compounded in highly populated metropolitan areas with large numbers of microwave ovens, Bluetooth devices, cordless phones, wireless game controllers, and other devices which occupy the ISM band. This contributes to an ever increasing need for effective interference mitigation schemes in Wi-Fi networks.

Wideband Interference: Microwave Oven. The most common source of interference in the ISM band is the microwave oven (MWO). Nearly every home, apartment, office, and restaurant in a large metropolitan area contains an MWO, and its operation can severely degrade WiFi transmission. Both residential and commercial MWOs are characterized by a wide-band frequency profile. However, the interference varies in time with a nearly even on-off cycle corresponding to its 60 Hz AC power supply. This corresponds to a period T of approximately 16.7 ms, a key characteristic used in both modeling and identifying MWO signals.

Narrowband Interference: Bluetooth. Other than MWOs, most interferers in the ISM band occupy a bandwidth smaller than the approximately 20 MHz WiFi channel. The majority of these devices employ a frequency-hopping spread spectrum (FHSS) method to randomly change between several carrier frequencies throughout the 2.4 GHz ISM band. Although FHSS devices should theoretically cause minimal interference with Wi-Fi communication, their increased prevalence has been shown to be detrimental to throughput in both simulation and field testing.

Efforts to eliminate the effects of these interferers on WiFi networks have been focused on providing robust, interferer-agnostic mitigation techniques. Previous work has focused on detecting and classifying signals, but alleviating their negative effects based on this classification has not yet been explored. For example, Airshark [1] and RFDump [2] act as low-cost spectrum analyzers that are helpful as network diagnostic tools. The authors of these papers mention the potential of interference mitigation but neglect to implement it. Other work has been done in avoiding interference at the MAC layer by switching to a different channel or changing the rate of transmission, but this approach does not adjust mitigation to best remedy the effects of a specific interferer. The disconnect between these techniques fundamentally limits the potential for interference mitigation.

1.1 iSCISM

Previous work has motivated the formation of the iSCISM project as a bridge between identification and mitigation. The process can be decomposed into three stages; the first of these is the detection stage, in which information is gathered regarding the signals present. The information required to properly identify a signal varies widely based on the signals being identified and the comparisons used.

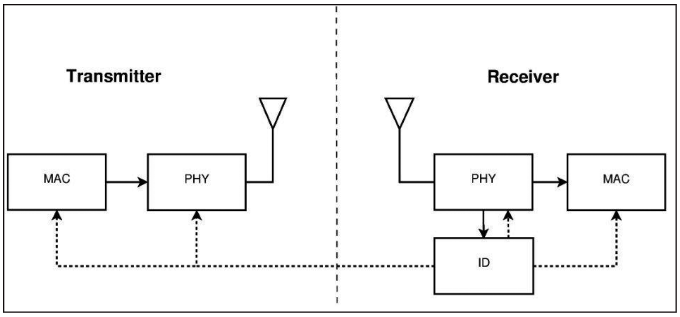


Figure 1• Block diagram of the iSCISM mitigation architecture.

Features such as changes in a received signal's phase or the times of transmission can be used to differentiate between different types of interferers. The block diagram of the iSCISM system is shown in Figure 1. The PHY layer extracts data from the received signals, an identification algorithm specifies the existence of interferers, and mitigation algorithms are activated to mitigate the effects of the interferers present in the system.

The second stage in the process of utilizing unique mitigation schemes is specifying the type of interference present in the relevant frequency band. Even after selecting a specific parameter upon which to base identification, the methods for implementing this stage can vary widely. Decisions may be based simply on hard-coded threshold techniques or on more complicated methods such as machine learning. The former keeps computation low while the latter may permit higher degrees of accuracy.

The third stage is implementation of the appropriate interference mitigation algorithm, and this will determine how practical the first two prove. Mitigation schemes are the most pivotal portion of the project as a whole. By combining interference identification and previously developed mitigation schemes to produce an interferer-specific mitigation architecture, the coexistence of Wi-Fi with various interferers in the ISM band can be better facilitated. The schemes to be implemented, however, must be both computationally efficient and effective in alleviating the effects of interferers. Several mitigation schemes were tested and analyzed for these factors.

Varying the operation of mitigation schemes based on detected interferers requires a modular and quickly adaptable system. Software-defined radio (SDR) presents an excellent solution. Advances in SDR have enabled transmitters and receivers to become reconfigurable through software instead of dedicating excess hardware to perform particular tasks. The notion of re-configurability is exploited in order to create systems that are able to adapt in real-time to environmental changes. Through the application of SDR, cognitive radio communication systems are able to most efficiently allocate resources based on information about their surroundings, and secondary users are able to share spectrum with primary users. These developments make SDR the ideal foundation on which to construct an interference specific mitigation architecture.

The rest of the paper is outlined as follows: The formation of a testbench in MATLAB® is described in Section 2. This served as the platform for the simulations completed thus far. Section 3 details the signal identification techniques implemented in this testbench. The techniques for peak detection and specific algorithms for identification are detailed in Section 4. Mitigation schemes are described in Section 5. The theory behind each is explained and a brief analysis is offered. The experimental procedure of implementing peak detection and identification algorithms on the MATLAB testbench as well as the results obtained from the testbench are detailed in Section 6, and Section 7 outlines the conclusions drawn from the completed work and the direction for future work.

2 . MATLAB Testbench

To properly test the performance of iSCISM in improving throughput in the presence of interferers, two platforms are used. First, a MATLAB simulation serves as the modular testbench on which several identification and mitigation schemes are tested. This allows blocks of code to be interchanged easily, and both recorded and simulated interference signals may be used for testing. Once results are obtained and the most successful algorithms are selected, the iSCISM system will be implemented on an SDR platform for testing in a more realistic setting.

The MATLAB testbench has been developed to provide a way to test several different interference mitigation schemes for a variety of interference signals, signal to noise ratio (SNR) and signal to interference ratio (SIR) values, and time durations. While MATLAB is often used for Physical (PHY) layer simulations, discrete event simulators such as OPNET® and ns-3 are typically used for Medium Access Control (MAC) layer simulations. These do not allow manipulation or observation of the PHY layer, so MATLAB was chosen as a simulation medium. Due to computational restrictions in MATLAB and the relatively low sampling rate of the USRP2, it is feasible to simulate only a portion of one 20 MHz channel. Hopping across multiple channels is not a necessary design feature, so this should be sufficient for modeling the Wi-Fi component of the testbench.

A limited MAC layer is implemented with the following procedure. To begin, input data are split into packets. At each iteration of a loop, a packet is passed through an 802.11g PHY layer simulation. After filtering with a Rayleigh fading channel, the signal power is calculated. Then, additive white Gaussian noise and interference are added at the packet level to ensure constant SNR and SIR regardless of input signal amplitude. Instead of computing checksums, the receiver compares the received, demodulated bits to the original packet and the variable ack is set to 0 if there are any errors. If ack = 1, then throughput is incremented based on the current bitrate and the next packet is sent on the following iteration.

Prior to each transmission, a wait period is allocated to account for delays between consecutive transmissions. These time delays represent the Short Interframe Space (SIFS), Distributed Interframe Space (DIFS), and exponential backoff times specified in the 802.11g protocol. Interference signals corresponding to these time frames are selected and processed to ensure that the relevant data are extracted for use in the identification process. This allows for processing of interference signals in both the transmission and wait periods. To keep track of elapsed time over several iterations, the variable **T** is incremented based on the number of OFDM symbols transmitted and the duration of the waiting periods.

Within this testbench framework, the effects of simulated and actual interference signals have been determined and mitigation solutions have been evaluated based on throughput and bit error rate. Because the system is not yet mapped to a real-time hardware implementation, computational complexity has been assessed by monitoring the amount of time required to conduct simulations.

3. Feature Extraction

Gathering information from the iSCISM system's received signals relies almost exclusively on peak detection. The peak detection algorithm is therefore the most pivotal point of any identification algorithm. Successful identification of interference, regardless of the type of interference or method used, first requires accurately distinguishing between background noise and signals of interest.

The edge detection algorithm is an adaptation of the algorithm used in the RFDump architecture [2], but filtering techniques are refined to produce more computationally efficient and effective results. The algorithm averages in time and downsamples the received signal by means of a cascaded integrator and comb (CIC) filter. Downsampling the received signal allows for longer averaging times with minimal increases in computational complexity. A 50 μs time average is used as it effectively removes noise without blurring the often tightly-spaced peaks. A sloped filter identical in length to the averaging filter is then applied. This produces a time-averaged derivative of the received signal. Sharp peaks in the resulting signal can then be easily identified, and a threshold is used for separating out these peaks. The value for this threshold is set based on the maximum value of the time-averaged signal. Times at which the derivative surpasses this threshold are labeled as rising edges, and times at which the derivative falls below the negative of the threshold are labeled as falling edges.

Following peak detection, feature extraction is a simple task. The phase of an interferer as it changes in time can be observed through a simple arctangent operation. The time stamps exported with the peak data can be easily manipulated to analyze the timing characteristics of the received signals. It should be noted, however, that this approach is done separately from spectral analysis. Analyzing the frequencies of interfering signals in any detail requires similar peak detection algorithms in the frequency domain. Analyzing the spectrum in a way that provides sufficient detail for interference identification requires more complicated time and hardware intensive architectures. While it is likely a promising method for identifying signals, the relative complexity dictates its exclusion from this discussion.

4. Classification Algorithms

Machine learning algorithms can be used to classify interfering signals based on their unique features. This section outlines the theory behind supervised learning algorithms, where *supervised* refers to training prior to classification using signals that are mapped with certainty to particular devices.

K-Nearest Neighbors The K-nearest neighbors (KNN) algorithm has been used in the past to classify biological signals immersed in noise [3]. This idea could easily be extended to interference identification in Wi-Fi networks, such as in the iSCISM system. The KNN algorithm classifies training samples according to their Euclidean distance to other samples in the feature space. This is done by a probability density estimate in the form where $p(x) = \frac{K}{NV}$ where $p(x)$ is the unknown probability density at sample x, V is the volume of the region containing x, N is the total number of sample points, and K is the number of points in the region containing x. The K -nearest neighbors algorithm fixes K and determines the value of V from the data by centering a sphere at point x and increasing the radius until K points are inside the sphere. This approach is applied to each class, and samples are assigned based on the class the majority of their K -nearest neighbors are a part of. The posterior probability of class membership is given by

$$(C_k|x) = \frac{p(x|Ck)}{p(x)} = \frac{K_k}{K} \qquad [4].$$

Naive Bayes The Naive Bayes classifier has been used to classify network traffic and obtain high accuracy [5]. This classifier could also be used for interference identification in Wi-Fi networks as interference is also network traffic. The Naive Bayes classifier assumes that classifications of an object are independent from all other classifications of other objects, and features are also assumed to be independent from all other features.

According to Bayes' Theorem and conditional independence assumptions, the joint probability of class variables and feature variables can be rewritten as the product of the class prior $p\ (C_k)$ and all i independent distributions $P\ (f_i | C_k)$ where f_i are the feature variables. Maximum likelihood is then used to match an object to a class. A maximum a posteriori (MAP) classifier is similar to a Naive Bayes classifier but instead of assuming independent features, it uses the MAP decision rule and picks the most probable hypothesis.

Support Vector Machines A binary SVM classifier composes a hyperplane in a feature space of a higher dimension than the original sample space so that data can be linearly separable. A *margin* is the shortest (and therefore perpendicular) distance from a hyperplane to data points, and the hyperplane where the margin is maximized coincides with minimum error and is therefore a useful decision boundary. For a problem in the form $y(x) = w^T \phi(x)+b$ where $\phi(x)$ is the feature space transformation, b is the bias term, w is the vector normal to the hyperplane and the sign of $y(x)$ is what classifies the N input data vectors $x_1, ..., x_N$, the decision hyperplane is represented as

$t_n(w^T \phi(x_n) + b) \geq 1\ \forall\ n = 1, ..., N$; these are constraints to the quadratic programming problem

$$\mathrm{argmin}_{w,b} \frac{1}{2} \|w\|^2$$

which can be solved using Lagrange multipliers $a_n \geq 0$. This problem becomes one in which we minimize with respect to w and b and maximize with respect to a. After the model has been trained, $y(x)$ can be expressed as

$$\sum_{n=1}^{N} a_n t_n k(x, x_n) + b \text{ where } k(x, x') = \varphi(x)^T \varphi(x')$$

is the kernel function. After satisfying the Kuhn-Tucker conditions, either $a_n = 0$ or $t_n y(x_n) = 1$ for every data point. The data points for which $a_n = 0$ do not contribute to new points being classified, so only the data points for which $t_n y(x_n) = 1$ remain. These points are *support vectors* that lie on the maximum-margin hyperplane. This hyperplane is the decision boundary used for binary classification [4].

5. Interference Mitigation

To be applicable to the iSCISM system, interference mitigation solutions should satisfy the following criteria:

Noncollaborative: In a collaborative environment, one device can coordinate its transmissions as to not interfere with those from another device. Collaboration is certainly possible in the context of a multistandard wireless device containing two or more collocated radios. Coexistence among multiple cognitive RF devices would simplify the problem greatly, but collaboration depends on the exchange of information which cannot be assumed in general. A purely receiver side solution is also preferable to one that requires communication or synchronization between the transmitter and the receiver.

Low layer design: In order to be implemented on a Wi-Fi card, the solution must be mapped to hardware or low level software. Information should be extracted from waveforms or bitstreams rather than packets. Techniques are considered only if they employ either the PHY layer or the MAC sublayer. Hybrid, cross-layer (MAC/PHY) approaches are also permissible.

Scalability to multiple networks: An ideal solution will have no negative consequences if all radios in a large urban area adopt the technique. Therefore it must avoid a tragedy of the commons in which the system's benefits vanish once it is implemented in a majority of Wi-Fi networks.

A sensing and interference mitigation approach has several advantages and disadvantages. The obvious benefit of software-defined radios adapting to interference is increased throughput between the transmitter and receiver. This method also provides better utilization of RF spectrum, a limited resource for wireless devices. With different coexistence strategies for different interferers, the solution should perform better than any individual approach. However, the potential for error increases with complexity. Latency and other forms of overhead make interference mitigation inappropriate for all cases. An unrobust implementation also runs the risk of performing worse than a baseline Wi-Fi network without interference suppression.

Several methods for interference mitigation were considered, including frequency diversity [6], adaptive filtering [7, 8], timed transmission, and rate adaptation. The latter two techniques were selected for development and implementation the iSCISM project.

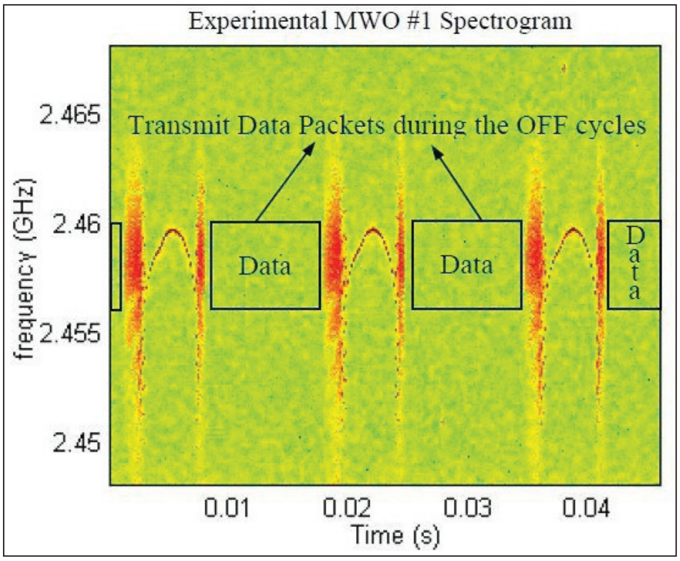
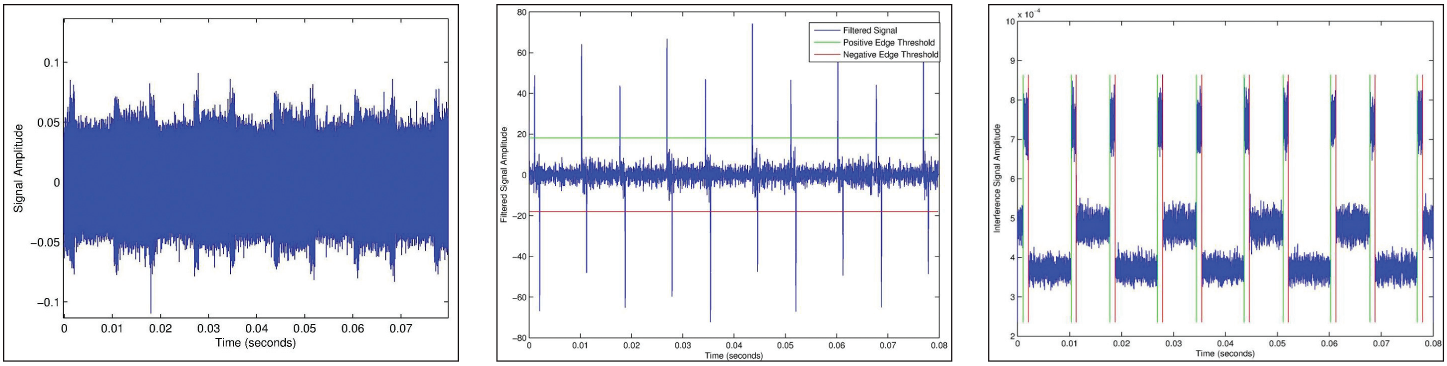


Figure 2 • MWO spectrogram with optimal transmission times. From (9).



(a) Received Signal (b) Differentiated Signal (c) Detected Edges

Figure 3 • The process of edge detection performed on a microwave oven is demonstrated above. The input signal is averaged, downsampled, and differentiated before a threshold separates peaks.

5.1 Timed Transmission

One approach to mitigating microwave oven interference is to simply avoid transmitting while the MWO is in its on cycle. Although this reduces the transfer rate to half, the bit error rate has been shown to drop to zero in experimental tests [9]. Transients are detected as on-off boundaries, and Figure 2 shows that the recommended transmit time is in between these bursts. Although this method forces the Wi-Fi transmitter to operate at a 50 percent duty cycle, the absence of MWO interference should result in greater overall throughput. While it is a promising PHY layer solution, it requires the transmitter to adjust its output based on the receiver's identification procedure.

5.2 MAC Rate Adaptation

The 802.11 Wi-Fi standard allows for several modulation schemes and transmit rates. One general strategy against interference is to dynamically adjust the rate in accordance with the presence of interference. Several common rate adaptation schemes assume that packet loss is due to multipath or fading channel conditions and neglect the possibility of RF interference. Algorithms such as the WOOF scheme [10] identify losses due to wireless network congestion and take no action instead of decreasing the data rate. Although this mitigation technique focuses on accommodating large, multi-user networks instead of mitigating external interference, it shows considerable improvement in throughput compared to naive methods which decrease the transmit rate in the event of any packet loss.

Previous efforts have attempted to detect and reduce microwave oven interference using the MAC sublayer by correlating the number of consecutive successful transmissions with an MWO on-off cycle [11]. Because the system only requires spectrum sensing and MAC characteristics to sense MWO interference, it should be easily implemented on commercial 802.11 radios without hardware modifications.

Once interferers have been detected, interference mitigation algorithms adjust MAC parameters to improve overall performance. One algorithm dynamically adjusts the contention window in order to transmit more often in the off cycle and less often in the on cycle. In some cases, an MWO may cause a radio to interpret the spectrum as busy even when it poses no threat of interference. One solution is to adjust the clear-channel-assessment threshold and transmit at the moderate, previously unacceptable SIR. However, these algorithms only work in specific cases based on the microwave oven's proximity to the transmitter and receiver.

Rate adaptation was determined to be a simple method for implementation against Bluetooth interference in the iSCISM system. The scheme presented in [12] was implemented: the rate was increased upon eight successful transmissions and decreased upon three failed transmissions.

6. Experimental Results

After preliminary results with simulated interference models, experimental data were obtained using a USRP2 software-defined radio. Complex valued, 32-bit floating-point interference signals for both microwave ovens and Bluetooth headsets were recorded at a sampling rate of 20 MS/s.

6.1 Peak Detection

The edge detection algorithm performed well at relatively low INRs. Testing was performed on simulated Bluetooth and MWO interferers. These were weighted and added to produce a single interference vector containing both interferers. Edges were noted as accurately detected if an edge was marked within 5 μs of the generated peak. At an INR of 2 dB for the simulated Bluetooth interferer and 1 dB for the simulated MWO interferer, 84% of the generated Bluetooth peaks and 90% of generated MWO peaks were located accurately. Minimal false positives were generated (less than one every 5 ms). These results at relatively low INR values demonstrate reasonable accuracy at non-ideal conditions. Higher INR values yield better detection accuracy with fewer false positives. At lower INRs the performance degrades, but such low-power interference signals do little more than raise the noise floor, and thus detection and identification of these signals is relatively unimportant. Successful implementation of the edge detection algorithm is demonstrated in Figure 3.

While these results refer to the performance of edge detection purely on simulated signals, similar results have been demonstrated when this algorithm is applied to recorded interferers. Quantifying the success of the algorithm on these interferers is much more difficult as there is no accurate definition of the location of transmission edges against which the identified edge times can be compared. The performance of the algorithm with recorded interferers was carefully observed over several different sets of data, and it also proved successful in locating these edges accurately. One notable difference, however, arose in the detection of microwave oven edge detection. While the overall envelope of the simulated MWO signal matched that of the recorded MWO signal very closely, the signal power rose much more gradually to peaks in power than in the simulated MWO. The resulting upward slope was much longer in duration than the simulated counterpart, and this caused the detection of multiple closely-spaced edges within a single upward slope. This inconsistency, however, has a negligible effect on the performance of the identification algorithm as the overall timing of an MWO interferer is maintained quite closely.

Algorithm	Training time(s)	Run time(s)	Accuracy, all BT	Accuracy, all MWO
KNN	0.01	0.011	93%	100%
Naive Bayes	0.04	0.008	97%	93%
MAP	0.02	0.011	97%	93%
SVM	0.09	0.012	96%	98%

Table 1 • Experimental Classifier Results.

6.2 Classification Results

Using a computer with an Intel Core i7 CPU 870 @ 2.93 GHz processor and 6 GB of RAM, the classifiers were trained and tested. Training data and test data came from two separate simulations of interference modeling and feature extraction. At a sampling rate of 20 MHz, duration of 60 ms, SIR of 10 dB, and SNR of 10 dB (INR = 0 dB), the run times of each classifier were averaged over ten trials. The training times are less relevant than the run times since training is a one-time expense and can be done offline without real-time concerns. The algorithms were each tested for accuracy by using test data that was comprised of entirely Bluetooth information or entirely microwave oven information, for the cases of only Bluetooth interference being present and only microwave oven interference being present. Accuracy measurements were averaged over ten trials for each interference type. These results are shown in Table 1. For the case of mixed BT and MWO interference, qualitative results showed roughly the appropriate amount of each interference type was classified correctly. However, since ground truth is unknown, quantitative results have yet to be established.

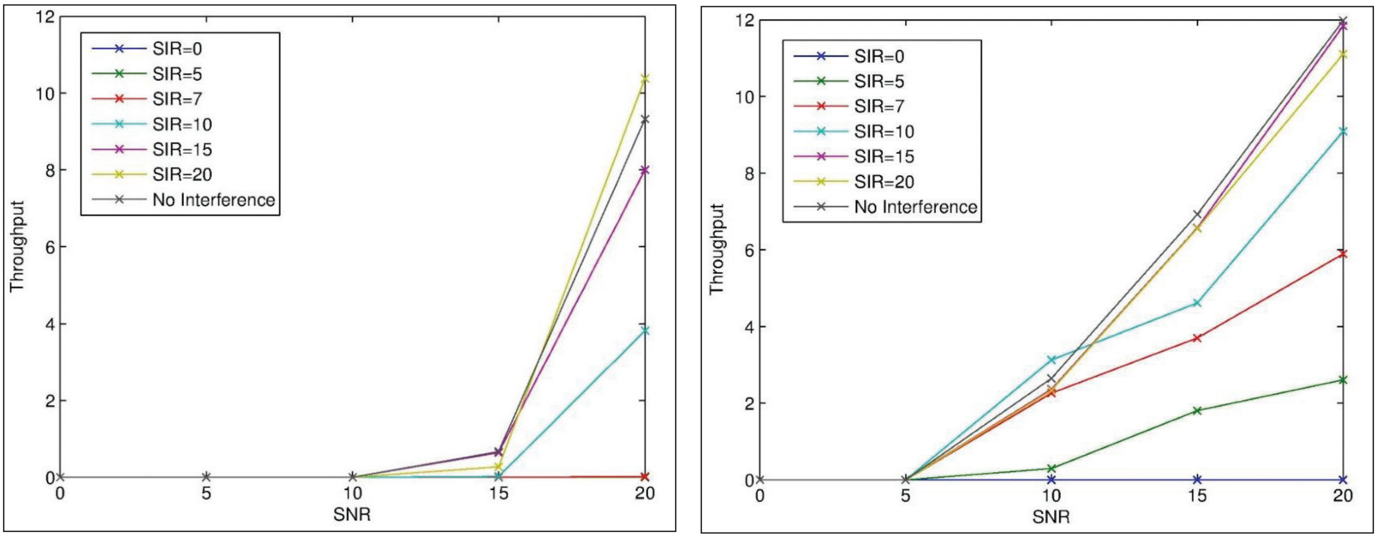


Figure 4 • 802.11 throughput (left) with experimental BT interference and (right) with experimental BT and rate adaptation.

6.3 Mitigation Results

Due to time constraints, tests of timed transmission performance in the presence of MWO interference were determined to be inconclusive. Rate adaptation was tested using two interference signals, each experimentally recorded from a different Bluetooth headset. In the adaptation scheme, data rate was permitted to vary among 6, 12, and 24 Mbit/s according to the scheme described in Section 5.2. This was then compared to a simulation with a fixed data rate of 24 Mbit/s for SIR and SNR values ranging from 0 dB to 20 dB. In addition to the bit error rate improving by an order of magnitude for rate adaptation at SNR values from 5 dB to 15 dB, the throughput increased for all SIR values above 5 dB SNR. While a normal 802.11g simulation with the second Bluetooth interferer yields no data transmission at 10 dB SNR, all but 0 dB SIR had nonzero throughput for the same SNR level when rate adaptation was applied. Figure 4 shows these results averaged over 20 iterations. At a constant SNR value of 20 dB, the throughput improvement ranged from 7% at 20 dB SIR to over 100% at 15 dB SIR.

7 . Conclusions

Edge detection, identification, and interferer-specific mitigation schemes have each been tested. The edge detection algorithm yields accurate detection of interference transmissions at relatively low INRs. Interference identification algorithms have been shown to accurately identify interferers based on the detected edges. Naive Bayes is the likely candidate for the final identification algorithm as it produces accurate results with the smallest computational cost. In addition, throughput improvements in a Wi-Fi system have been demonstrated when rate adaptation (for the case of a Bluetooth interferer) is used to mitigate the effect of interference.

7.1 Future Work

Some work must still be done to finalize the implementation of iSCISM on the MATLAB testbench. This includes incorporating the finalized identification and mitigation schemes onto one unified platform. The ability of the system to dynamically identify the presence of interferers can then be fully tested. Identification algorithms will be quantitatively evaluated further by examining performance when the interference is a weighted sum of both MWO and Bluetooth interference. In addition, timed transmission must be tested more extensively to determine the amount of improvement it provides.

The next version of iSCISM will be completed on a USRP2 SDR platform. To more accurately simulate the effects of interferers on a Wi-Fi system and iSCISM's ability to mitigate them, a Wi-Fi transmitter and receiver platform will be implemented through the GNU Radio interface. iSCISM will then be added to this platform. The computational simplicity of low-level design should permit the finalized version to then be used on a Wi-Fi card.

iSCISM's performance may also be improved when more identification and mitigation schemes can be implemented and tested on the SDR platform. Several promising but complex methods of interference identification and mitigation were not tested. Should these prove effective, they may be incorporated in future versions of the iSCISM platform.

Notes: MATLAB® is a registered trademark of The MathWorks, Inc. OPNET® is a registered trademark of OPNET Technologies, Inc.

References

[1] S. Rayanchu, A. Patro, and S. Banerjee, "Airshark: Detecting non-wifi rf devices using commodity wifi hardware," 2011, unpublished.

[2] K. Lakshminarayanan, S. Seshan, S. Sapra, and P. Steenkiste, "RFDump: An architecture for monitoring the wireless ether," in *Proc. 5th ACM International Conference on emerging Network EXperiments and Technologies (CoNEXT'09),* Rome, Italy, Dec. 1–4, 2009, pp.253–264.

[3] V. D. Gesu, G. L. Bosco, and L. Pinello, "A one class knn for signal identification: a biological case study," *International Journal of Knowledge Engineering and Soft Data Paradigms,* vol. 1, no. 4, pp. 376–389, 2009.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning.* New York, NY: Springer Science + Business Media, 2006.

[5] Y. Liu, Z. Li, S. Guo, and T. Feng, "Efficient, accurate internet traffic classification using discretization in naive bayes," in *IEEE International Conference on Networking, Sensing and Control,* 2008.

[6] K. Premkumar and S. Srinivasan, "Diversity techniques for interference mitigation between IEEE 802.11 WLANs and Bluetooth," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'05),* vol. 3, Berlin, Germany, Sep. 2005, pp. 1468–1472.

[7] A. Soltanian, R. E. V. Dyck, and O. Rebala, "Rejection of bluetooth interference in 802.11 WLANs," in *Proc. IEEE 56th Vehicular Technology Conference (VTC'02),* vol. 2, Birmingham, AL, May 2002, pp. 932–936.

[8] Z. Zeng, B. Allen, and A. Aghvami, "Performance evaluation of a bluetooth interference canceller in IEEE802.11b wireless networks," *IEEE Trans. Consum. Electron.,* vol. 51, no. 4, pp. 1188–1196, Nov. 2005.

[9] T. Taher, M. Misurac, J. LoCicero, and D. Ucci, "Microwave oven signal interference mitigation for Wi-Fi communication systems," in *5th IEEE Consumer Communications and Networking Conference (CCNC'08),* Las Vegas, NV, Jan. 2008, pp. 67–68.

[10] P. A. K. Acharya, A. Sharma, E. M. Belding, K. C. Almeroth, and K. Papagiannaki, "Congestion-aware rate adaptation in wireless networks: A measurement-driven approach," in *5th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08),* San Francisco, CA, Jun. 16-20, 2008, pp. 1–9.

[11] G. Li, S. Srikanteswara, and C. Maciocco, "Spectrum-sensing based interference mitigation for WLAN devices," in *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08),* Bangalore, India, Jan. 5–10, 2008, pp. 402–408.

[12] S. Miyamoto, S. Harada, and N. Morinaga, "Performance of 2.4GHz-band wireless LAN system using orthogonal frequency division multiplexing scheme under microwave oven noise environment," in *IEEE International Symposium on Electromagnetic Compatibility, (EMC'05),* vol. 1, Aug. 8–12, 2005, pp. 157–162.

About the Authors:

Joe Baylon, Ethan Elenberg, and Samantha Massengill will soon graduate from the EE program at The Cooper Union for the Advancement of Science and Art, New York, NY.