A ROBUST APPLICATION DETECTOR FOR INTELLIGENT WIRELESS COLLABORATION

Kevin Pietsch *Sean Mason*

Lockheed Martin Advanced Technology Labs
Cherry Hill, New Jersey 08002
Email: {kevin.pietsch, sean.mason}@lmco.com

ABSTRACT

We present an approach for detecting application level protocols over a wireless communications link, without the need for demodulation or decryption. Our detector is suitable for diverse radio types, since only simple external signal features are used as inputs. We show that the Profile Hidden Markov Model (PHMM) is well suited to this task, due to the probabilistic nature of the wireless channel and the discrete nature of application level traffic. We include results evaluating the detection performance for two application protocols in 802.11 in the presence of background traffic. Using only inter-arrival time and packet size as inputs, we show capable performance for both detection and discrimination between the two. We go on to argue that our approach will work with other physical layer and medium access control layer types due to the simplicity of the inputs.

Index Terms— Wireless collaboration, Spectrum Inference, Application Detection, Spectrum Sharing

1. INTRODUCTION

The continuing exponential increase in wireless radio frequency traffic, which is expected to grow from 4.4 exabytes per month globally in 2015 to an excess of 30 exabytes per month in 2020 [1] will drive increasing attention to the problem of spectrum sharing. Over the past decade, methods for radios to find and utilize clear spectrum have been a key focus of the burgeoning field of cognitive radio research which seeks to mitigate this impending crisis [2, 3, 4]. Unfortunately this sense-and-avoid approach that focuses on finding unused spectrum can only do so much in the face of such a massive increase in wireless traffic. We propose a new approach that emphasizes intelligent collaboration, in which radios devise methods to optimally share channels to ensure valuable capacity does not go unused. One key technical hurdle will be enabling devices to collaborate to share the wireless medium despite having no direct means of communication.

A major focus will be to develop methods by which radios infer the state, behavior, and intentions of others using only externally observable features. The process for radio collaboration over a shared wireless medium can be broken into a few steps, as outlined in Fig. 1. To start an agent selects a channel that, despite being in use by an incumbent, has some perceived tolerance to additional traffic. The agent may select its transmission parameters according to that estimate before entering the channel. Once sharing the channel, the agent will continue monitoring the state of the incumbent (state estimation) and adjust its parameters accordingly to maintain efficient channel usage by the incumbent. State estimation can include but is not limited to incumbent health monitoring (e.g. throughput estimation), association of incumbent behavior changes with agent actions, and detection of critical application level protocols.

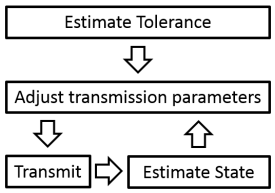

Fig. 1. General outline of the collaborative spectrum sharing process.

The focus of this paper is a novel inference method to support cognitive radio collaboration. We present a detector for application level protocols that works without demodulation, making it useful for radios that do not have a common communications protocol. This allows for the possibility of defining spectrum sharing rules based on application type, rather than radio network type, which represents a shift from standard cognitive radio research.

Our detector models the sequential flow of features of the incumbent's signaling and attempts to associate that model with stored models of application layer protocols. We choose the profile hidden Markov model (PHMM) [5, 6] as our detector for its proven ability to detect genetic sequences with mutations. This is a crucial property, since interleaving with random amounts of background traffic, and erasures due to signal fades and packet collisions–close analogies of genetic mutations–are unavoidable in the wireless domain.

The rest of this paper is organized as follows. In Sec-

DISTRIBUTION A. Approved for public release: distribution unlimited. This research was developed partly under funding from the Defense Advanced Research Projects Agency's (DARPA) MTO Office under contract HR0011-11-C-0033. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

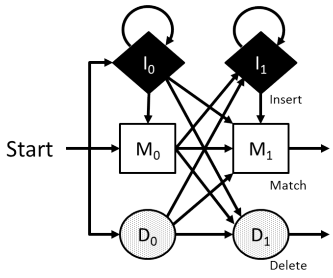

Fig. 2. The Profile Hidden Markov Model with Insert (I), Match (M), and Delete (D) states.

tion 2 we describe this paper's contribution and describe related work. In Section 3 we formulate the problem and introduce the detector. Section 4 describes our experimental approach for validation with real data and Sections 5 and 6 contain our results and a discussion of them, respectively. Section 7 describes a live demonstration of this technology given at the 2015 Defense Advanced Research Projects Agency's (DARPA's) Wait, What Conference, and we conclude in Section 8.

2. CONTRIBUTIONS AND RELATED WORK

This paper investigates the PHMM as a novel detector for known applications over an unknown wireless network. We argue for the PHMM's suitability to the wireless domain by demonstrating its robustness to interleaved traffic streams, packet collisions, and missed detections. Further, we argue for its suitability to intelligent collaboration applications due to its ability to extract meaning from an incumbent network's signaling without needing demodulation.

An analogous application of the PHMM in the wireline domain can be found in [7]. Here the PHMM is investigated for its ability to perform sequence detection in encrypted data due to its ability to handle variations between the observed sequences of packets across multiple realizations of the same protocol. While certain properties of the wireless domain have somewhat close analogies to the wireline domain (e.g. insertions due to multiplexed traffic), the wireless domain provides enough unique challenges (e.g. noisy features and signal fades) to warrant its own study.

3. TECHNICAL APPROACH

A key enabling characteristic of wireless communications is that the unobservable temporal characteristics of application layer protocols manifest as observable temporal characteristics of physical layer (PHY) signaling. A model of the relationship between application and PHY is depicted in the top half of Fig. 3. Top rectangle: we model the application layer protocol we wish to detect as a series of discrete states

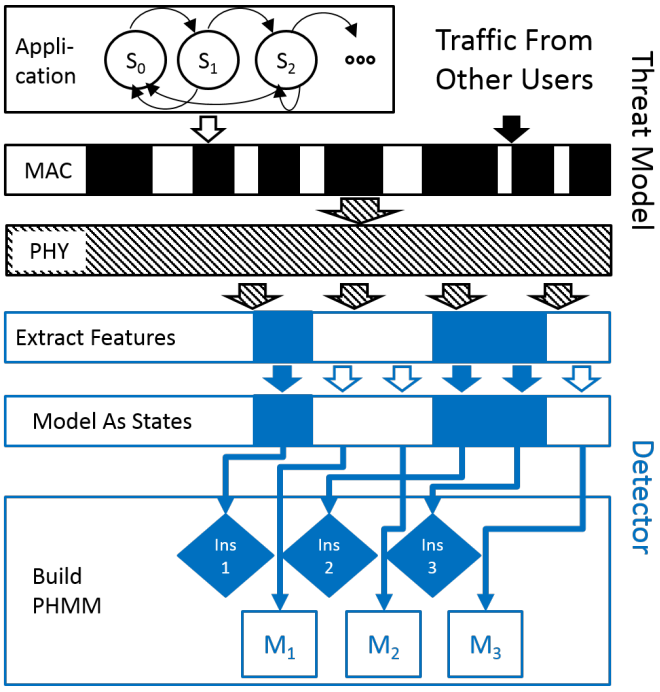

Fig. 3. A diagram of our network behavioral model along with a description of our training process. White rectangles and boxes represent data belonging to the sequence of interest. Blue and black shapes indicate background traffic and striped patterns indicate a mix.

$S = S_0, S_1, \cdots, S_{N-1}$ and the transition probabilities between them (i.e. as a Markov process). Second rectangle down: at the medium access control (MAC) layer the application is mapped to physical signaling and potentially interleaved with background traffic (i.e. traffic belonging to other users). This is depicted as white and black rectangles, respectively. To an observer the presence of background manifests itself as random amounts of insertions between states of the sequence of interest. Third rectangle down: at the PHY the combined traffic is transmitted. This is depicted as a striped pattern to indicate a mixture of traffic pertaining to the sequence of interest and background.

The bottom half of Fig. 3 guides the description of both the training and detection process for the PHMM. In both cases this requires reducing the raw measurements made on the PHY (i.e. digital inphase-and-quadrature samples) in dimensionality for a PHMM to be useful. This is the combined function of the blocks labeled "Extract Features" and "Model as States", and is presented in more detail in Fig. 4. Feature extraction is run either at fixed time and frequency intervals or on identifiable bursts of data. The input can be described as a received signal $y(t \in T_i)$ defined by time extent T_i, and the output is a vector of feature values $\Phi_i = [\phi_1,\phi_2,...,\phi_K]$. Next a clustering algorithm is used to map each Φ_i to one of M clusters $[C_1,C_2,...,C_M]$ where M is usually chosen to be on

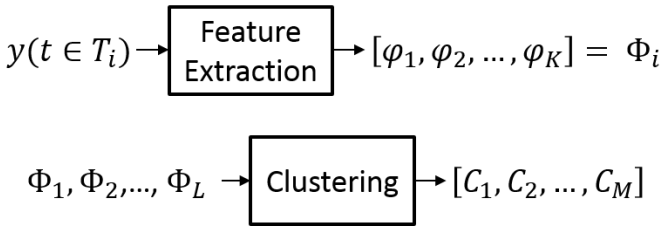

Fig. 4. Feature extraction and clustering maps raw measurements of the physical layer to a finite set of clusters.

the order of 10 (which is much less than the number of unique feature vectors). States are then defined by which cluster they belong to $(1, 2, ..., M)$. The relatively low number of clusters is necessary for presenting the PHMM with a finite number of possible states in order to have meaningful state transition properties.

3.1. Training

Training is done in a supervised manner, and begins by recording multiple instances of signaling containing the sequence of interest. The output of training are two models: the "match" that models the sequence of interest S, denoted as λ_S, and the "background" that models background traffic, denoted as λ_B. The background model represents the overall distribution of features corresponding to packets that do not carry the sequence of interest and is created using a single state HMM.

The PHMM model λ_S allows for variations between the collected sequences of interest by defining two additional states for each position in the standard Markov chain (see Fig. 2). Insert states (denoted by I_k) allow for one or more extra packets to be inserted between match states. Delete states (denoted by D_k) allow for the k^{th} match state (M_k) to be skipped.

The training process concludes by labeling the training corpus according to what step of the sequence of interest each state pertains, or as insertions. States that pertain to the sequence of interest are labeled as match states, and ones that pertain to other traffic are labeled as insertions. Deletions are specified when a state is missing. Transition probabilities are obtained empirically from the training corpus.

3.2. Detection

The first step in detection is feature extraction, where the raw received signal is converted to feature vectors, i.e. $y(t \in T_i) \rightarrow \Phi_i$. Next for each Φ_i the detector scores the match between it and each cluster from the trained model $P(\Phi_i|C_1)$, $P(\Phi_i|C_2)$,...,$P(\Phi_i|C_M)$. This results in a match score for every cluster at every timestep which is then used to evaluate the likelihood that the observations are due to λ_S, $P(S|\lambda_S)$. This quantity can be evaluated using the forward algorithm, a standard dynamic programming method. More details about sequence detection using the forward algorithm can be found in [6].

Detection is declared when the ratio of log likelihood of model match to background match exceeds a threshold, Γ:

$$\frac{\log\left(P(S|\lambda_S)\right)}{\log\left(P\left(S|\lambda_B\right)\right)} > \Gamma. \tag{1}$$

4. EXPERIMENTAL SETUP

4.1. Test Protocols

We evaluate our detector against two application layer protocols: network association using dynamic host control protocol (DHCP) [8] and the secure socket layer (SSL) handshake [9]. Both protocols are used ubiquitously for wireless networking and detection of either creates a more detailed understanding of the current state and function of a network. Testing was done via a leave-one-out approach: $L+1$ instances of a given protocol were collected and $m(L+1)$ tests were run where m represents the number of Monte Carlo trials. For each group of m tests, one instance was chosen to be detected against, and the remaining L instances were used to train a model.

4.1.1. Network Association

Network association includes all traffic involved with associating a host to the wireless network including obtaining an IP address from the access point via DHCP. We recorded network association data during the formation of an 802.11 network.

4.1.2. SSL Handshake

The SSL handshake precedes encrypted traffic and is a feature of hyper text transfer protocol secure (HTTPS). HTTPS is an extension of normal HTTP where the SSL handshake is added as a precursor to the transfer of data to set up an encrypted connection between host and server. We visited the following websites to collect implementations of the SSL handshake: https://google.com, https://yahoo.com, and https://apple.com.

4.2. Test Data Collection

For PHY/MAC we used IEEE 802.11g [10]. We collected data over the air in an office environment during normal business hours on 802.11 channel 1 (2.412 GHz) in order to ensure a realistic concentration of background traffic. We chose channel 1 after Wireshark–an open source network packet analyzer [11]–logs showed it to have the highest arrival rate of background packets among all available 802.11 channels. Three nodes were used: the server, host, and passive observer. A high level diagram of this setup is presented in Fig. 5. A description of each element follows:

The server was an 802.11g access point connected to the internet via a wired connection. This served as the hub of a

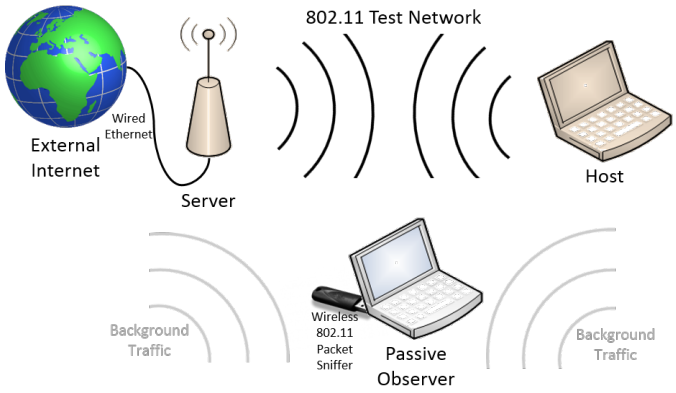

Fig. 5. Diagram of experimental setup. All curved lines represent wireless 802.11 traffic on channel 1. Black lines represent traffic belonging to the test network, while gray ones represent traffic emanating from background emitters.

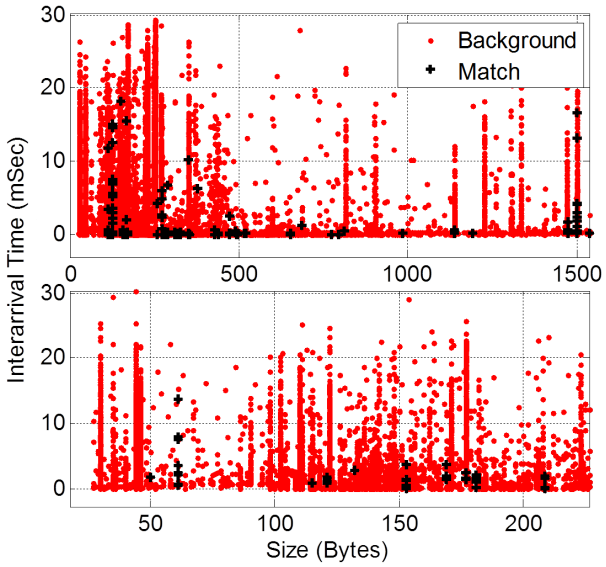

Fig. 6. Features corresponding to match states and background states for the SSL handshake (top panel) and network connect (bottom). The horizontal axes are constrained to the span of match states. No noise was added to either feature for this plot.

private 802.11 network to which only the Host could join, on channel 1. This network is herein referred as Test Network. The host was a laptop which connected to the Test Network via internal WiFi adapter. Collects were performed by initiating the sequence of interest from this laptop over the Test Network. The passive observer was a separate PC running Wireshark to monitor all traffic on 802.11 channel 1 using a specialized USB adaptor. Despite the passive observer's ability to demodulate WiFi, the Wireshark logs it produced were only used to generate features indicative of the observable properties of the signalling. A more thorough explain our features is in the next subsection.

Collection followed these steps: 1) Begin Wireshark packet capture on Passive Observer. 2) Generate protocol traffic using Host PC. 3) Stop Wireshark packet capture. 4) Disconnect Host from Test Network.

4.3. Feature Set

We used two features: packet interarrival time and packet size in bytes. Reliable measurement of interarrival time amounts to the detection of signal presence with a sufficient time resolution, meaning the feature extractor we require is no different than one required by many primary user sensing studies in mainstream cognitive radio research, including ones which rely on the HMM such as [12] and [13]. Packet size can be approximated as a function of the detected modulation type used and packet duration. Modulation detection for digital signals has been under consideration for years (see [14]), and while recognition of the modulation of orthogonal frequency division multiplexing (OFDM) subcarriers is not as mature as modulation recognition for single carrier systems, both classification of OFDM versus single carrier modulation and blind estimation of OFDM symbol timing–two key capabilities that could be combined for this task–are understood (for an example of each see [15], and [16], respectively).

For each sequence S a PHMM was trained by defining match states by fitting features pertaining to each state of S to a cluster (making one cluster per match state), and insert states by fitting all data collected between match states to a single cluster. The background single-state HMM was created by defining one background cluster pertaining to traffic collected when S was not present. Each cluster was defined as a Gaussin mixture model (GMM), i.e. parameterized by the mean, variance, and mixing coefficients of a mixture of Gaussian functions. The clustering algorithm used was expectation maximization (EM). For a description of formulation of the GMM using EM see [17], chapter 9. The transition probabilities between states was determined by counting and averaging transitions between states in the training data.

5. EXPERIMENTAL RESULTS

To state the necessity for a sequential detector like the PHMM we plot the two features (interarrival time and packet size) on the same axes in Fig. 6. From these figures it is clear that separation of these features with a non-temporal classifier would require a decision boundary defined by an extremely high number of parameters. Due to the small amount of training data, there would be an extremly high likelihood of errors due to overfitting.

Our data set included 24 SSL handshake messages, 10 network connects, and 21 background collects of approxi-

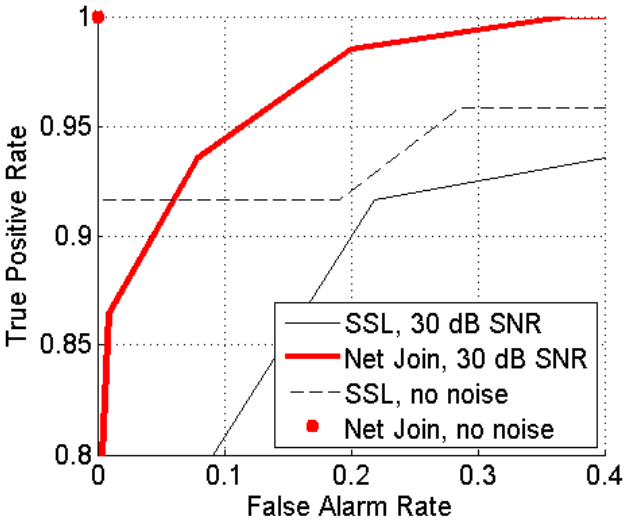

Fig. 7. Probability of detection vs. probability of false alarm for SSL handshake "SSL" and Network connect with DHCP "Net Join", for noiseless features and features in 30 dB of SNR. Note that the same values for Γ were used for all four cases.

mately 30 second duration each. We evaluated the detection performance for each model versus background for both noiseless and noisy features where noisy features were subjected to additive Gaussian noise with 30 dB SNR. In each noisy case results were averaged over 10 Monte Carlo trials. The probability of false alarm versus true positive rate for all four cases (Network Connect and SSL, noiseless and 30dB SNR) is given in Fig. 7.

We also include results on the cross-detection performance of our system by evaluating the positive detection rate of each model against the other in Fig. 8.

6. DISCUSSION

Two possible reasons why the Network Connect detector performs better than the SSL handshake detector in Fig. 7 are the difference in feature values (apparent in Fig. 6) and differing sequence lengths: the SSL model had 8 states while the Network Connect model had 9.

The cross detection results in Fig. 8 affirm that a system employing both detectors simultaneously can gain a clear understanding of the state of the incumbent network at the application layer. For either traffic type the rate at which the correct detector rang up was up to 90 percent higher than the rate at which the incorrect detector did.

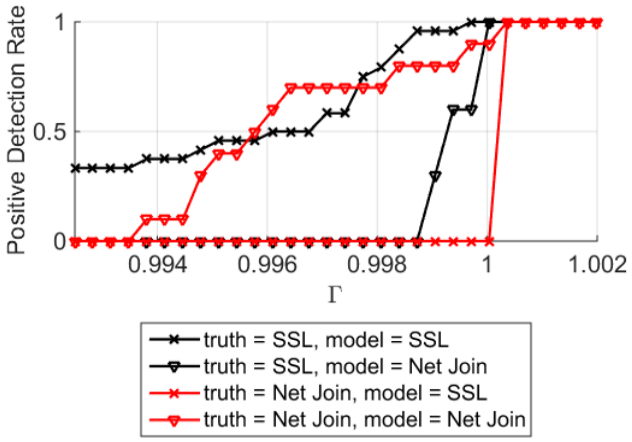

Fig. 8. Positive detection rate (the percentage of instances in which detection was declared) for SSL handshake "SSL" and Network Join with DHCP "Net Join" for both types of detector. In the legend, truth indicates the type of application being hosted. Model indicates which detector was employed.

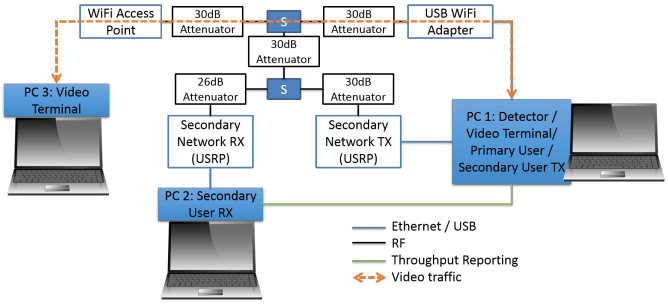

Fig. 9. Connection diagram for the demonstration setup at DARPA Wait, What 2015.

7. DEMONSTRATION AT DARPA WAIT, WHAT?

A real time demonstration was presented at the DARPA Wait, What? Forum in September 2015 [18]. A secondary network used an application detector to determine whether an incumbent 802.11 network was hosting a Google Video chat session or an FTP file transfer, and tuned its transmission parameters accordingly before entering the channel. Fig. 9 is a diagram of the physical setup.

8. CONCLUSION

This paper presents a detector for application layer protocols hosted on a wireless communications link that does not require a detailed understanding of the physical layer or medium access protocol in use (i.e. no need for demodulation). To evaluate this detector's suitability to the task of spectrum sharing, particularly for estimation of an incumbent's state, we showed reliable detection performance for two ubiquitous net-

working applications.

Future work will focus on increasing the level of autonomy in collaborative radios, allowing them to learn and adapt to the environment as well as understand the behaviors of other networks. Two possible frontiers include methods for jointly estimating the state of an incumbent and adjusting transmission parameters, and methods that require less pre-training for robust state and tolerance estimation.

Acknowledgment

We thank Mr. Paul Tilghman for his valuable contributions and Dr. David Dorsey and Mr. Thomas Szumowski for their valuable insights. We also thank Mr. Kevin Rigney and Mr. Matthew Zimmerman for their contributions to the demonstration at the DARPA Wait, What? forum.

9. REFERENCES

[1] "Cisco visual networking index," *Web page: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf*, 2015.

[2] Ayaz Ahmad, Sadiq Ahmad, Mubashir Husain Rehmani, and Naveed Ul Hassan, "A survey on radio resource allocation in cognitive radio sensor networks," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 2, pp. 888–917, 2015.

[3] Ekram Hossain, Dusit Niyato, and Dong In Kim, "Evolution and future trends of research in cognitive radio: a contemporary survey," *Wireless Communications and Mobile Computing*, vol. 15, no. 11, pp. 1530–1564, 2015.

[4] Shree Krishna Sharma, Tadilo Endeshaw Bogale, Symeon Chatzinotas, Bjorn Ottersten, Long Bao Le, and Xianbin Wang, "Cognitive radio techniques under practical imperfections: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 4, pp. 1858–1884, 2015.

[5] Sean R. Eddy, "Profile hidden markov models.," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.

[6] Richard Durbin, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge university press, 1998.

[7] Charles V Wright, Fabian Monrose, and Gerald M Masson, "On inferring application protocol behaviors in encrypted network traffic," *The Journal of Machine Learning Research*, vol. 7, pp. 2745–2769, 2006.

[8] Ralph E Droms and Ted Lemon, *The DHCP Handbook*, Pearson Education, 2002.

[9] Alan Freier, Philip Karlton, and Paul Kocher, "The secure sockets layer (SSL) protocol version 3.0," 2011.

[10] IEEE 802.11 Working Group, "IEEE standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements–part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std*, vol. 802, Mar 2012.

[11] Gerald Combs, "Wireshark-network protocol analyzer," *Web page: http://www. wireshark. org/*.

[12] Zhanwei Sun, Glenn J Bradford, and J Nicholas Laneman, "Sequence detection algorithms for PHY-layer sensing in dynamic spectrum access networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 1, pp. 97–109, 2011.

[13] Chittabrata Ghosh, Carlos Cordeiro, Dharma P Agrawal, and M Bhaskara Rao, "Markov chain existence and hidden markov models in spectrum sensing," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–6.

[14] Octavia A Dobre, Ali Abdi, Yeheskel Bar-Ness, and Wei Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *Communications, IET*, vol. 1, no. 2, pp. 137–156, 2007.

[15] Domenico Grimaldi, Sergio Rapuano, and Luca De Vito, "An automatic digital modulation classifier for measurement on telecommunication networks," *Instrumentation and Measurement, IEEE Transactions on*, vol. 56, no. 5, pp. 1711–1720, 2007.

[16] Helmut Bolcskei, "Blind estimation of symbol timing and carrier frequency offset in wireless OFDM systems," *Communications, IEEE Transactions on*, vol. 49, no. 6, pp. 988–999, 2001.

[17] Christopher M Bishop et al., *Pattern recognition and machine learning*, vol. 1, springer New York, 2006.

[18] "DARPA Wait, What? Future Technology Forum, September 9-11, 2015," *Web page: http://www.darpa.mil/news-events/wait-what*.