Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 495070, 12 pages
doi:10.1155/2008/495070

Research Article

Passive Classification of Wireless NICs during Rate Switching

Cherita L. Corbett,[1] Raheem A. Beyah,[2] and John A. Copeland[3]

[1] *Computer and Network Security Group, Sandia National Laboratories, 7011 East Avenue, MS 9011, Livermore, CA 94550, USA*
[2] *Communications Assurance and Performance Group, Department of Computer Science, Georgia State University, 34 Peachtree Street, Suite 1451, Atlanta, GA 30303, USA*
[3] *Communications Systems Center, School of Electrical and Computer Engineering, Georgia Institute of Technology, 75 Fifth Street, Atlanta, CA 30308, USA*

Correspondence should be addressed to Raheem A. Beyah, rbeyah@cs.gsu.edu

Received 22 August 2007; Accepted 27 November 2007

Recommended by Lin Cai

Computer networks have become increasingly ubiquitous. However, with the increase in networked applications, there has also been an increase in difficulty to manage and secure these networks. The proliferation of 802.11 wireless networks has heightened this problem by extending networks beyond physical boundaries. We propose the use of spectral analysis to identify the type of wireless network interface card (NIC). This mechanism can be applied to support the detection of unauthorized systems that use NICs which are different from that of a legitimate system. We focus on rate switching, a vaguely specified mechanism required by the 802.11 standard that is implemented in the hardware and software of the wireless NIC. We show that the implementation of this function influences the transmission patterns of a wireless stream, which are observable through traffic analysis. Our mechanism for NIC identification uses signal processing to analyze the periodicity embedded in the wireless traffic caused by rate switching. A stable spectral profile is created from the periodic components of the traffic and used for the identity of the wireless NIC. We show that we can distinguish between NICs manufactured by different vendors and NICs manufactured by the same vendor using their spectral profiles.

1. INTRODUCTION

Computer networks have become more ubiquitous. Furthermore, the world has come to rely on these networks to provide transport for many different mission critical services. The increasing number of networked applications makes it difficult for network administrators to control what traffic traverses their networks. This difficulty is largely a result of the user's ability to easily modify the medium access control (MAC) and internet protocol (IP) addresses forcing network administrators to install third-party software on machines they want to manage.This approach is not ideal for several reasons: (1) the software is usually costly and thus not an option for many organizations; (2) one must have cooperation from the node; and (3) additional software equates to additional opportunity for vulnerabilities.

Extending the boundaries of computer networks with the advent of 802.11 wireless networks further heightens the problem of managing the nodes and traffic on computer networks. Many institutions use application-layer authentication tied to campus (or corporate) IDs and passwords. The problem with only using this approach to authenticate users is that the credentials are not tightly coupled to an individual. That is, a valid user can easily give his/her credentials to another user who desires to access the network, or the user may simply be tricked into revealing his/her credentials via an advanced phishing technique. This has alarming implications that question the overall reliability of any security technique (including 802.11i) that depends on information supplied by the user.

In addition to the risk of users giving their access away to unauthorized individuals, there is also a significant concern of valid users, themselves, bringing harm to the network by introducing unauthorized machines. These unauthorized nodes may contain malicious processes that can harm the network or connected nodes. Some vendors [1–6] provide solutions to this problem but require cooperation from the end node. These solutions are costly and unlikely to be wholly adopted.

We propose the use of spectral analysis to identify the type of wireless network interface card (NIC). This mechanism can be applied to support the detection of unauthorized systems that use NICs that are different from that of a legitimate system.

Our approach to establishing the identity for different types of NICs focuses on the implementation of the rate-switching algorithm, a function required by the 802.11 standard. We show that differences in the implementation of this function cause unique traffic patterns that can be used to discern between NICs manufactured by different vendors and NICs manufactured by the same vendor. We motivate the need for signal processing and apply it to analyze the periodicity embedded in the wireless traffic caused by rate switching. A spectral profile is created from the periodic components of the traffic and used for the identity of the wireless NIC.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Section 3, we discuss the composition of a wireless NIC and present opportunities for distinguishing between different types of NICs. Section 4 presents the rationale for using signal processing and introduces our technique. We also discuss how to represent a wireless traffic stream as a signal and how to compare spectral content in Section 4. In Section 5, we present an empirical analysis of rate switching conducted at a local hotspot. Section 6 gives the spectral analysis techniques used to distinguish between NICs during rate switching, in addition to qualitative and quantitative results. Section 7 concludes the paper and discusses future work.

2. RELATED WORK

Despite new security enhancements, the risk of intrusion is still a legitimate concern because preventive measures may be circumvented, cost prohibitive, or not practiced at all. As a result, intrusion detection systems for wireless environments have emerged to detect unauthorized access. Detecting unauthorized access affords an opportunity to respond to the intrusion and curtail the potential damage to preserve the privacy and integrity of the network.

Wright [7] detects unauthorized access by identifying media access control (MAC) address spoofing. Many of the attack tools (e.g., FakeAP, AirJack, and Wellenreiter) aimed at obtaining unauthorized access rely on spoofing of the MAC address of an authorized access point or legitimate client. The technique used in [7] monitors the sequence number field on the 802.11 frame header as a parameter to characterize normal behavior of a wireless LAN. The sequence number field is a sequential counter that is incremented by one for each nonfragmented frame. An attack is identified by a large gap in sequence numbers for an active MAC address. Additionally, if the sequence value increments sequentially with a changing source MAC address, BSSID, and SSID values, the behavior is also considered an attack. This technique can be evaded if the attacker is able to set the sequence number field to an arbitrary value. The approach also fails if the attacker uses a MAC address (authentic or spoofed) that has not previously been seen on the network. In general, signature-based techniques, such as this, require a continuous update of attack signatures to stay current. Further, this approach is not effective against novel attacks.

Commercial products, such as ReefEdge [8], AirDefense [9], and AirMagnet [10], offer a more comprehensive security solution with services for performance monitoring, intrusion detection, policy monitoring, and intrusion protection. However, these current systems do not address stealthy intruders that do not exhibit anomalous behavior or generate a sequence of events matching the pattern of an attack. For example, a hacker may have obtained a user name and password from an authorized user via reconnaissance and phishing techniques. In which case, the attacker appears to have legitimate access, and it does not exhibit alarming behavior since it had the proper credentials.

An alternative method to defend a wireless network from unauthorized access is to establish an identity for legitimate systems. Normally, the MAC address of the network interface card (NIC) serves as a unique identifier. However, attackers can spoof the MAC address of legitimate devices. The following techniques attempt to create an identity for nodes.

WiMetrics [11] is a commercially available monitoring and intrusion protection system. It implements an identity profiling process that can preauthorize a user through a registration process or authorize a user on the fly by probing the wireless device to derive an identity profile based on the response. Probing wireless stations is intrusive, and as the number of clients increases, the already constrained network becomes burdened with additional traffic imposed by the system. This approach has other drawbacks including the administrative overhead of the preauthorization process. In addition, a hacker could elude the system by crafting responses to the probe request to impersonate the identity of a legitimate user, reducing the effectiveness of this scheme.

IPass Inc. developed DeviceID [12], a software-based authentication technology. DeviceID creates a digital fingerprint using random segments of serial numbers from different hardware components within the device. It consists of two components, server and client software. The server encrypts and inventories the digital fingerprint in a database. The client resides on all end-point devices to establish secure sockets layer (SSL) connections for secure transmission of the device's fingerprint required for hardware authentication. This approach is intrusive and suffers from administrative overhead involved in distributing the client software and updating the database every time a hardware component changes in the device. Further, this approach generates traffic, placing additional strain on the wireless link.

Radio frequency fingerprinting captures the unique characteristics of the RF energy of a transceiver. When a radio transmitter is placed in transmit mode, a transient is generated by the frequency synthesizer whose function is to generate the carrier frequency used for transmission. It has been determined that the turn-on transients generated are distinct enough that positive identification of the transmitter is possible. This technology was originally used in the cellular industry to identify fraudulent clones [13]. Researchers at Carleton University [14] have extended this approach to control access amongst BlueTooth wireless devices with future plans to include 802.11 transceivers. To implement this technology in a wireless LAN, special equipment for processing RF signals would be required at each access point. The cost of new equipment can become prohibitive especially for large networks with many access points. This was not of significant concern to the cellular industry because each tower services thousands of subscribers dissipating the cost of the equipment.

Kohno et al. [15] demonstrate a method for remotely fingerprinting a physical device by exploiting the implementation of the TCP protocol stack. When the TCP timestamp option is enabled, outgoing TCP packets reveal information about the sender's internal clock. The authors' technique exploits microscopic deviations in the clock skews to derive a clock-cycle pattern as the identity for a device. For machines that do not enable the timestamp option by default, such as those running Windows 2000 and Windows XP, this approach becomes an active one. In such case, the active fingerprinting technique initiates a connection and tricks the fingerprintee into using the timestamp option. The active approach must violate the TCP specification in order to execute the trick. The drawback to the active technique is that it is detectable to the fingerprinted device. Furthermore, the entire approach only applies to TCP traffic and can be evaded by spoofing the TCP timestamp field or setting it to an arbitrary value.

In [16], we present an initial approach at using spectral analysis to distinguish between network cards during rate switching. Though this work proved effective, it was limited in that it focused on a limited number of network cards, the experiments were conducted in a controlled environment, and we were only able to accurately classify cards when they generated user datagram protocol (UDP) constant bit-rate (CBR) flows. The controlled environment was an isolated area with negligible interference from external wireless nodes where we used an artificial noise source for interference to trigger rate switching on and off in a controlled manner. We improve our initial work by detecting additional cards, modifying our technique such that it is transport layer protocol independent, and classifying cards in a real setting with random interference.

3. WIRELESS NETWORK INTERFACE CARD

A wireless network interface card (NIC) is installed into a host to carry out the physical transmission of a packet over the air waves. To do so, the IEEE 802.11 specification requires the implementation of two layers: the physical (PHY) layer and the medium access control (MAC) layer. To support this implementation, the NIC is organized into hardware, firmware, driver software, and utility software. The functions of 802.11 PHY are entirely implemented in hardware. The firmware is a microprogram semipermanently embedded into ROM to control the hardware. It works to communicate between the hardware and driver software. Driver software accepts generic I/O commands from the OS of the host and then converts them into instructions that the device can understand. The utility software is used to configure parameters to change the overall behavior of the hardware and software. The 802.11 MAC is implemented by a combination of hardware and software. The exact split is vendor-specific and greatly impacts the performance of the NIC.

3.1. *Opportunities for distinction*

The IEEE 802.11 standard specifies services that a wireless NIC must provide. However, the standard does not dictate how some of these services are to be implemented. It is left to the interpretation of the card manufacturer as to how to implement the 802.11 standard. We focus on the ambiguities in the 802.11 standard to differentiate between NICs manufactured by different vendors. This is analogous to operating system fingerprinting [17, 18], which exploits differences in the implementation of the TCP/IP protocol stack to determine the type of an operating system.

To support the transmission of data packets and to cope with the changing conditions of a wireless environment, the 802.11 standard engages services such as packet fragmentation, packet retransmission, adjusting transmission rates, reserving the link, probing network for connectivity, polling for packets to conserve power, and so forth. These services wield a certain behavior on the communication stream. This affords an opportunity to analyze properties about the stream, such as regularity in arrival rates and interarrival times of packets of different types and sizes. In addition to the basic services specified in the 802.11 standard, manufacturers often include acceleration hardware and software to increase performance gains and to support future standards prior to ratification. Enhancement techniques currently deployed to improve data transmission rates include data compression, frame bursting, overhead management, and client-to-client transfer [19]. Cards with different implementations of the 802.11 standard and with vendor-specific enhancements will have a different impact on the time-variant properties of a wireless stream. We exploit this fact to identify NICs manufactured by different vendors. Specifically, we hone in on the implementation of the rate-switching mechanism to distinguish between NICs.

3.1.1. *Rate switching*

Dynamic rate switching [20] is an important function of the 802.11 MAC and PHY. The 802.11 PHY has multiple data transfer rate capabilities that allow wireless cards to perform dynamic rate switching with the objective of improving performance. For example, 802.11b supports data transfer rates of 1, 2, 5.5, and 11 Mbps. Each rate corresponds to a different PHY modulation scheme with its own tradeoff between data throughput and operating range. It is the responsibility of the rate-switching algorithm to select the proper rate (modulation scheme) per packet that gives maximum throughput for certain link conditions.

Implementation of the rate-switching algorithm is vaguely specified by the 802.11 standard leading to vendor-specific solutions. Considering the sensitivity of intellectual property, we cannot guarantee completeness of the review of rate-switching algorithms. However, the literature survey [21–24] reveals three general approaches. Current WLAN products are believed to use a statistics-based feedback approach to rate-switching algorithms. Frame-error rate, achieved throughput, and acknowledged transmissions are used to estimate the quality of the link to select the appropriate rate for subsequent packet transmissions. The statistics-based approaches are outlined as follows.

Throughput-based approach

In throughput-based rate switching, a constant fraction of data is sent at two adjacent rates to the current rate. At the end of a specified window of time, the throughput at all rates is calculated as the ratio of number of bytes transmitted to cumulative transmission time. A switch is made to the rate that provides the highest throughput during the decision window. It is speculated that the Atheros WLAN card uses this approach in the Windows OS driver for 802.11a products based on the AR5000 chipset.

Frame-error rate

The frame-error rate (FER) algorithm computes the ratio of received acknowledgement frames to the number of transmitted data frames during a specified window of time. If the ratio exceeds some threshold and the current rate is not the minimal one, then a switch is made to the next lower rate. Conversely, if the FER ratio is below a second threshold, probe the link at the adjacent higher rate with n frames. If all n frames sent at that probing rate are acknowledged, then switch to that rate. The length of the window and thresholds control the behavior of the FER control algorithm.

Autorate fallback algorithm

The autorate fallback algorithm (ARF) is similar to FER in that it depends on acknowledged (ACK) frames. The protocol specifies that if ACKs for n consecutive data frames are not received by sender, then switch transmission rates to the next lower data rate and start a timer. If m consecutive ACKs are received, raise the transmission rate to the next higher data rate and cancel timer. Otherwise, when the timer expires, raise the transmission rate. If an ACK is not received for the very next frame, lower the rate again and reset the timer. The ARF algorithm is believed to be used in Lucent's 802.11 WaveLan II card.

Retry-based approach

The retry-based approach to rate switching is broken into two stages to address short-term and long-term variations in channel conditions. For the first stage, there is an ordered set of 4 pairs of rate transmission count parameters $\{r_0, c_0\}$, $\{r_1, c_1\}$, $\{r_2, c_2\}$, $\{r_3, c_3\}$. Transmission of data starts at rate r_0. If transmission fails, data is resent with rate r_0 for c_0-1 times. If transmission continues to fail, the algorithm tries rate r_1, c_1 times, then rate r_2, c_2 times, and finally rate r_3, c_3 times. Transmission is abandoned when the transmission has failed $c_0 + c_1 + c_2 + c_3$ times. In the second stage, the values for the 4 pairs of parameters are changed at a regular fixed interval. The rate r_3 is always chosen to be the minimum available rate. The rate r_0 is determined from previous values of r_0 and the transmission results over an elapsed period of time. Rates r_1 and r_2 are determined by r_0.

Summary

The rate switching algorithms described above each have a slightly different approach to estimating the channel quality and selecting a transmission rate. Each approach has a unique set of tunable parameters (i.e., thresholds for losses, successes, and retries, duration of timers, etc.) that control how the algorithms respond to transient and long-term changes in the condition of the wireless channel. Thus, each will have a different impact on transmission duration, interpacket delay, throughput capacity, occurrence of retransmissions, and other observable traffic characteristics. Additionally, differences in the setting of parameters within a particular algorithm will also display a distinguishable behavior.

All algorithms will have some form of the approaches highlighted above where they process frame transmission history to make a decision to switch rates. The agility of the rate-switching algorithm determines how often the rate changes and to what rate it changes, which impacts the traffic pattern. Some algorithms may send more frames at a higher rate and fewer frames at a lower rate, or vice versa. The selection of the transmission rate can widen or reduce the interarrival time between data frames. The side effects of the rate-switching algorithm not only impact the individual flow, but its effect is amplified within aggregate traffic as well. Research [25] has shown that the aggregate throughput declines and jitter and delay are compromised if at least one node in the network is performing rate switching. We investigate the periodicity imposed on the wireless traffic by the rate-switching algorithm. We look for distinctive spectral features to identify cards with different implementations of the rate-switching algorithm.

4. SIGNAL PROCESSING

The objective of our research is to show that it is possible to establish the type of wireless NIC by analyzing the temporal behavior of a wireless traffic stream. To achieve this objective, we need an extensive level of detail about the dynamics of a wireless stream. In this section, we present the rationale for applying spectral analysis, discuss signal representation of wireless traffic, explain the signal processing technique we use, and discuss how to compare spectral contents.

4.1. Rationale for spectral analysis

To date, most of the analysis on the behavior of wireless networks has been geared towards understanding transmission rates, throughput, and makeup of the composition of wireless stream (i.e., control traffic versus data traffic). This type of analysis has been successfully done in the time domain using monitoring tools and network analyzers. However, we need a much more sophisticated technique to characterize time-variant details to capture the artifacts embedded in the stream such as those caused by vendor-specific implementations.

Spectral analysis is a valuable tool for extracting timing information that may not otherwise be conveyed in the time domain. Spectral analysis is particularly useful in extracting periodic phenomena from (noisy) signals, because it succinctly compares the interrelationship between all data points. In the context of a communication network, periodicity means that if we see a frame in the network, then it is likely that after a constant period of time, we will see another packet passing through the same point. Networks inherently exhibit periodicity due to underlying protocols, network components, or host machines.

Spectral analysis has been shown to work well with identifying minute changes in the temporal behavior of network traffic. The authors of [26] applied spectral analysis to distinguish between normal TCP traffic and denial-of-service (DoS) attack traffic in aggregated, high-volume traffic. Hussain et al. [27] extended this work and showed that spectral analysis was useful in detecting the variations in the spectral profile attack stream as the composition (i.e., CPU speed, operating system, host load, etc.) of the attack host changed. This improved classification of the attacker beyond the type of attack tool. Partridge et al. [28] applied spectral analysis to wireless networks in order to deconstruct the traffic stream into individual flows or sessions. Their results showed that they were able to successfully detect individual flows without hearing transmissions directly related to an individual flow. We use spectral analysis to reveal differences in wireless streams generated by NICs that have different implementations of the rate-switching mechanism.

4.2. Signal representation

In order to apply spectral analysis, we need to represent a wireless stream of traffic as a signal that is suitable for the target signal processing function. The frame transmission process that occurs in WLANs can be described as a discrete event x, that occurs as a function of time t, that is, $x(t)$. There are a multitude of time-varying signals that can be generated from WLAN traffic. Even with encrypted traffic, the 802.11 header offers a rich source of information for signal representation: size of frame, type of frame, direction of frame, duration of frame, transmission rate of frame, received signal strength of frame, and so forth.

Once the information has been chosen to be represented by the signal $x(t)$, the signal must be uniformly sampled. A general approach is to pick an appropriate interval T, bin time into increments at that interval (nT), and count the number of events that arrive during that bin of time $(t, t+T]$:

$$x(t) = x(nT), \quad n = 0, 1, 2, 3, \ldots. \tag{1}$$

The evenly spaced time interval T is called the sampling interval of the signal. The sampling frequency F_s is its reciprocal $(F_s = 1/T)$.

To determine the sampling frequency, the Shannon sampling theorem [29] states that to reproduce a signal with its highest frequency component $F_{\max}$, the sampling frequency F_s must be at least twice $F_{\max}$. This frequency is referred to as the Nyquist frequency F_c $(F_c \geq 2 \times F_{\max})$.

4.3. Power spectrum density

A common spectral analysis technique is the periodogram [30], or power spectrum density (PSD). A PSD captures the power or spectral density a signal has over a range of frequencies. The magnitude of the power indicates the amount of regularity of the periodicity at the corresponding frequency. For our encoded wireless traffic signal, the PSD captures the periodicity in the arrival rate of frames. The magnitude corresponds to how often the arrival pattern occurs. PSDs are useful for identifying key frequencies to characterize the temporal behavior of a wireless stream.

4.3.1. Theoretical description

A PSD compares the interrelationships within a signal. It does so by using the discrete Fourier transform (DFT) of the samples of a signal and taking the squared magnitude of the result. The PSD P_{xx} of a signal of length L is given in

$$\hat{P}_{xx}(f) = \frac{|X_L(f)|^2}{f_s L}, \tag{2}$$

where the discrete Fourier transform (DFT) X_L is given in

$$X_L(f) = \sum_{n=0}^{L-1} x_L[n] e^{-2\pi j f n / f_s}, \tag{3}$$

and $x[n]$ is a discrete sequence of events.

The DFT takes a time-series representation of a signal and maps it into a frequency spectrum. It is a decomposition of a function into harmonics of different frequencies.

4.3.2. Welch method

During our analysis, we used the Welch average periodogram method (provided by the MATLAB signal processing toolbox) to estimate the power spectrum density. The Welch method [31] is implemented as follows.

(i) The input signal vector x is divided into k overlapping segments according to segment length l and number of overlapping samples *noverlap*.

(ii) The specified windowing function w is applied to each segment of x.

(iii) An *nfft*-point FFT is applied to the windowed data.

(iv) The modified periodogram of each windowed segment is computed.

(v) The set of modified periodograms is averaged to form the spectrum estimate $\hat{P}_{xx}(f)$.

(vi) The resulting spectrum estimate is scaled to compute the power spectral density as $\hat{P}_{xx}(f)/F_s$, where F_s is the sampling frequency.

```
(1) $[P_{xx}, \text{Freq}] = \text{PSD}(x)$              Estimate the PSD of time series $x(t)$, which returns a vector of frequencies Freq
                                          and a vector of power $P_{xx}$ that corresponds to the frequencies.
(2) $[\text{sorted\_}P_{xx}, \text{IX}] = \text{sort}(P_{xx})$        Sort the $P_{xx}$ vector in descending order, which returns an array of indices IX
                                          of the elements in $P_{xx}$ in descending order.
(3) [sorted_Freq] = Freq(IX)              Use the indices of IX to match the ordering with the sorted vector of power sorted_$P_{xx}$.
(4) spectral profile = sorted Freq(1 : $N$)   Use the first $N$ values of the sorted frequency vector to constitute the spectral profile.
```

Algorithm 1: Algorithm for comparing spectra.

The number of segments k into which x is divided is calculated as

$$k = \frac{(m - o)}{(l - o)}. \tag{4}$$

In this equation, m is the length of the signal vector x, o is the number of overlapping samples (*noverlap*), and l is the length of each segment.

The Welch method returns the PSD vector and the corresponding vector of frequencies. This is a measure of exactly what frequencies are present and at what magnitude. Averaging done in the Welch method reduces the influence of noise. Additionally, the smoothing done by the windowing function w reduces spectral background noise and clutter levels at the cost of some smearing of the peak energies in the frequency domain.

The Welch method depends upon several parameters: type of windowing function, segment size, number of data points to overlap between consecutive segments, and number of points for the FFT (*nfft*). The setting of these parameters affects the outcome of the Welch estimator.

4.4. Comparing spectra

The PSD estimator generates a spectrum of *nfft*/2 data points. It contains the magnitude of power at frequencies that are present in a signal. Ideally, one would like to use the complete spectra for comparisons between different signals. However, this can be computationally expensive. Rather than using the complete spectra, we select a subset of PSD values to represent the key spectral features of the signal using the algorithm shown in Algorithm 1. The algorithm below locates N frequency points that exhibit the greatest amount of power to constitute a spectral profile $F = \{f_1, f_2, f_3, \dots, f_N\}$. These key frequency points estimate the most prevalent arrival rates of frames in a wireless stream.

5. NIC IDENTIFICATION USING RATE SWITCHING

We have pinpointed rate switching as an IEEE 802.11 function that can be used to identify NICs (see Section 3.1.1). A rate-switching algorithm dynamically adapts the transmission rate per packet, based on the channel conditions to optimize performance. Additionally, the implementation of the rate-switching algorithm dictates the number of frames to transmit at the selected rate, how often to change rates, and the order in which the transmission rate is selected. This directly impacts the periodicity of a wireless stream. We use signal processing to extract the spectral content imposed by the rate-switching algorithm to identify wireless network interface cards.

5.1. Empirical analysis of rate switching

We conducted an empirical analysis to characterize the rate-switching phenomenon. We demonstrate, with measurements taken at a wireless hotspot, that rate switching occurs with reasonably high frequency and is also more likely to happen the longer an NIC has been transmitting.

5.1.1. Experimental setup

Analysis was conducted at a local hotspot on the campus of the Georgia Institute of Technology. Over the course of 7 days, we captured all traffic on the wireless network. We used a Toshiba laptop with a Linksys WPC11 wireless card to collect traffic. We put the wireless card into monitor mode using the *wlanctl-ng* utility [32] and stored the captured traffic using *tcpdump* [33]. With the card in monitor mode, we were able to detect the transmission rate associated with each packet collected, while *tcpdump* appended a timestamp to each packet. We used the timing information and transmission rate to generate statistics. Traffic was collected for a total of 13.3 hours over the course of 7 days. During our observation period, there were a total of 61 wireless clients that visited the hotspot.

5.1.2. Results

The results of our analysis show that rate switching is common at the hotspot. While this is definitely true for the hotspot we monitored, it is likely that RF interference occurs at most hotspots. Therefore, rate switching is likely a widespread, common phenomenon. Figure 1 shows the transmission rate of each data frame transmitted by one of the clients at the local hotspot. This particular client had undergone 279 changes to its transmission rate. Overall, Figure 2 shows how often rate switching occurred for all wireless clients over the entire observation period. Figure 3 shows that 67% of the clients performed rate switching, while 33% did not switch rates. Out of the clients that did not perform rate switching, 85% sent less than 9 packets (see Figure 4). If we exclude the nonswitching clients that sent less than 9 packets (assuming that these clients were never properly authenticated to the network), the percent of clients that performed rate switching becomes 92% (see Figure 5).

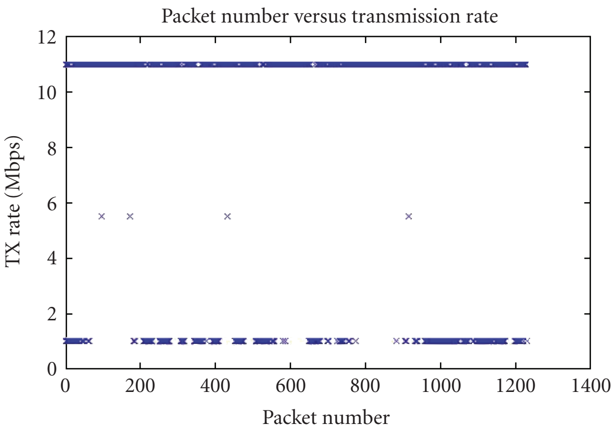


Figure 1: Host at local hotspot invoking rate switching.

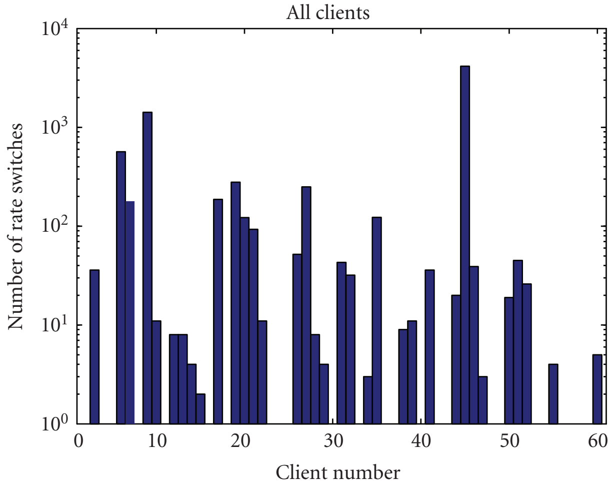


Figure 2: Number of rate switches for each client.

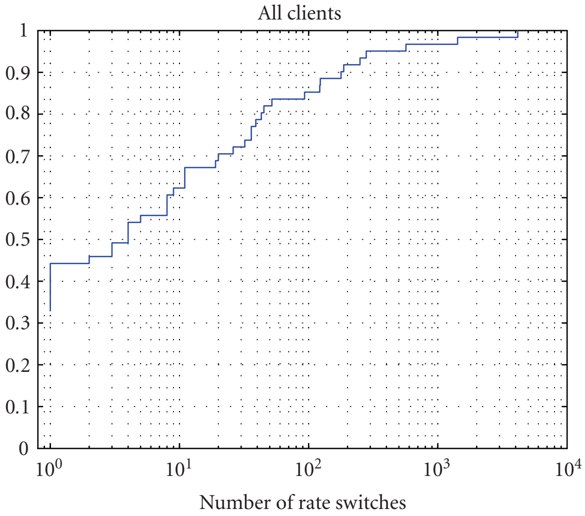


Figure 3: CDF of the number of rate switches for all clients.

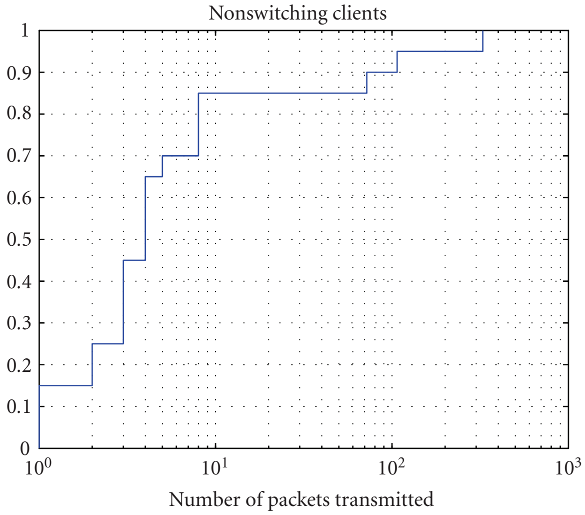


Figure 4: CDF of the number of packets transmitted by clients that did not perform rate switching.

Examining only the wireless clients that applied rate switching, Figure 6 shows that 90% transmitted more than 37 packets and 88% were connected to hotspot for more than 2 minutes. Also, 85% of these clients switched rates within the first 3 minutes of their connection.

We conclude that rate switching is a phenomenon that occurs. Our results show that the longer a wireless client is connected to the network and the more packets it transmits, the more likely rate switching is to occur. Therefore, rate switching is a viable attribute within the wireless NIC for distinguishing between cards.

6. CLASSIFYING HOSTS IN A REAL ENVIRONMENT

We have shown that wireless clients frequently exercise rate switching and that it is feasible to extract differentiating spectral characteristics as a result of rate switching for the identification of an NIC. In this section, we conduct experiments to show that rate switching is a stable attribute for identifying NICs in a real environment. Unlike the controlled environment [16], this environment consists of multiple heterogeneous clients contending for the network, multiple access points connected to the Internet, clients entering and leaving the network, and various user applications traversing the network. In addition to comparing cards by different manufacturers, we evaluate the differences and similarities within the same manufacturer. We also examine the effects of different higher-layer protocols on the spectral profile for rate switching.

6.1. *Experimental setup*

The following experiments were conducted at a hotspot on the campus of the Georgia Institute of Technology. Our wireless client was a 1 GHz Toshiba laptop with Redhat 9.0 that transmitted data to a remote node connected to the Internet (see Figure 7). We extended the controlled experiments [16] to include 2 additional Linksys WPC11 cards and an additional Lucent Orinoco/Gold card for a total of 6 NICs. (The cards identified in this section as Linksys3, Lucent1,

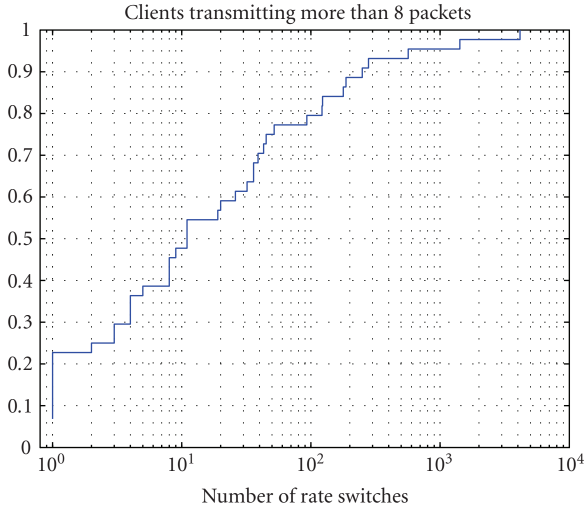


Figure 5: CDF of the number of rate switches excluding nonswitching clients that transmitted less than 9 packets.

and Dlink were the same cards used during the controlled experiments.) The Dlink and Linksys cards used the *prism2_cs* driver software, and the Lucent cards used the *orinoco_cs* driver software. For each card, we generated three separate traffic flows: one UDP flow and two TCP flows. The UDP traffic was generated using the *sock* program to transmit a 1470-byte packet every 5 milliseconds. Each UDP flow lasted approximately 10 minutes. To generate the TCP traffic, the client used FTP to upload a 164 MB file to the remote node. The FTP sessions lasted 9–14 minutes. During the experiments with the Linksys3 card, the card would often change wireless channels during the FTP session. As a result, the traffic captures were 1 to 2 minutes long.

A second laptop, placed next to the client, was used to collect traffic at the hotspot. We collected traffic using *tcpdump*. We used the *wlanctl-ng* utility which allowed the sniffer to prepend a *prism* header to the 802.11 frame while collecting traffic. The *prism* header allowed us to obtain the transmission rate and other physical layer information for each packet.

6.2. Analysis

For each of the experiments, we extracted the transmission rate and time stamps associated with data frames sent by our client to the access point. We used the transmission rate information to calculate the number of rate changes that occurred during each session. We found that all cards exercised rate switching while streaming traffic for all flows. Figure 8 shows the wireless network interface cards invoking rate switching (in the time domain) during UDP session. Similar graphs were generated for all other traffic flows, but they are not shown due to space constraints. Though it is possible to observe the rate-switching phenomenon in the time domain, it is difficult to extract a definitive recurring pattern that can be used to distinguish between cards. For this reason, we employ spectral analysis.

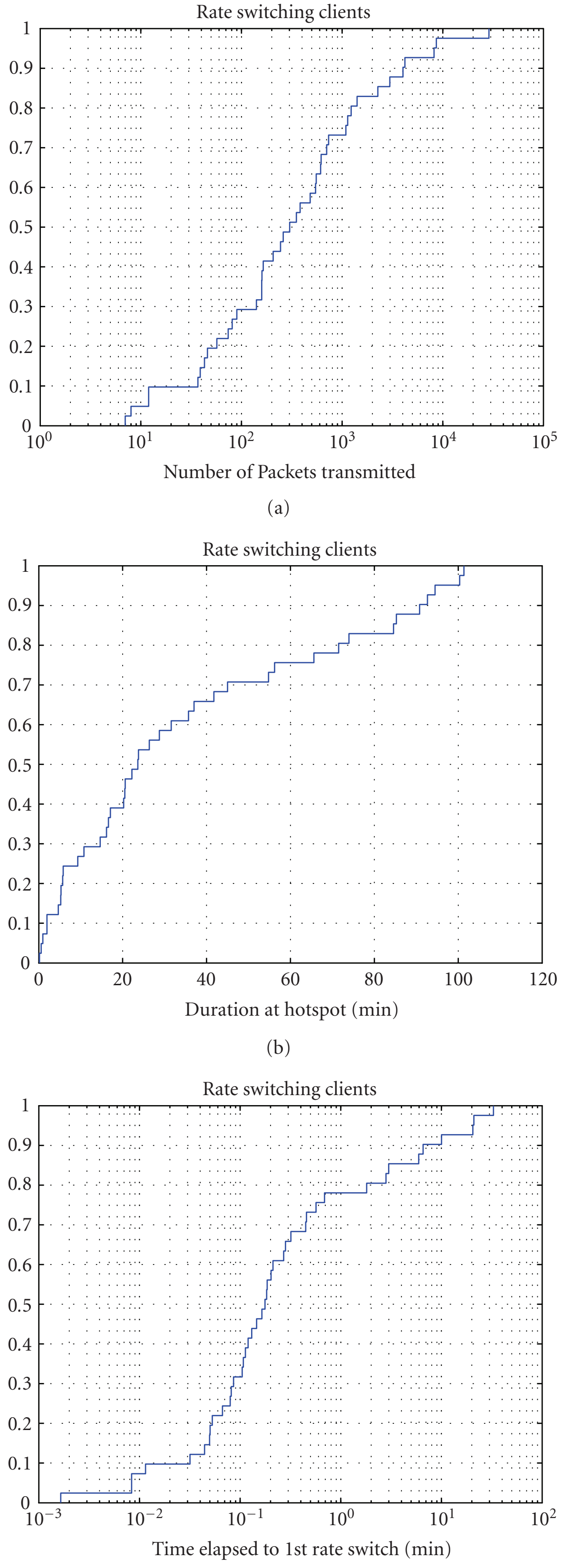


Figure 6: Rate switching clients: (a) CDF of the number of packets transmitted; (b) CDF of duration at hotspot; (c) CDF of time elapsed at 1st rate switch.

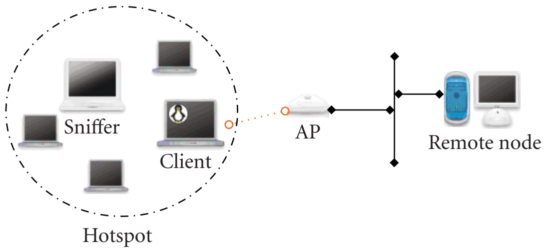


Figure 7: Experimental setup.

Table 1: Percent of segments matching F_R.

	UDP	TCP1	TCP2
Dlink	87.5	90	—
Linksys1	83.3	100	100
Linksys2	100	100	100
Linksys3	75	n/a[1]	n/a[1]
Lucent1	100	100	100
Lucent2	100	100	100

[1] The length of the traffic captures was too short to have enough segments for an accurate comparison.

Table 2: Classification of wireless cards.

Class 1	Class 2	Class 3
Dlink Linksys1 Linksys2	Linksys3	Lucent1 Lucent2

To create the time series of events, we used the data frames and corresponding timestamps. The time series was sampled at a rate of 600 Hz, counting the number of data frames sent in each 0.0167-second bin. The sampled signal representing each flow was partitioned into 60-second segments. Then, we estimated the PSD for each segment of the flow independently. To calculate the PSD, we used a 1024-point FFT; we sectioned the signal using 1024 data points with 50% overlap between sections, and used the Hanning windowing function. Figure 9 plots the PSD results for the first three segments of the TCP-flow1 for the Lucent1 NIC. Similar graphs were generated for all experiments, but they are not shown due to space constraints. Across all cards and flow types, visually comparing the spectra among different segments within the same traffic flow revealed similarities. This indicates that the impact rate switching has on traffic throughout the succession of the flow is steady.

To quantitatively compare spectra between segments of traffic within the same flow, we obtained the top 50 frequencies for each 60-second segment to represent the spectral profile $F_{\text{seg}} = \{f_1, f_2, f_3, \dots, f_{50}\}$ using the approach outlined in Section 4.4. Figure 10 illustrates the top 50 frequencies for the Lucent1 TCP-flow1. While only using a fraction of the values from the PSD estimate, the formation of a straight line near 0 Hz, 8 Hz, 16 Hz, and across the range of 212–239 Hz illustrates the fact that spectral content for different segments is similar.

Next, we randomly selected the spectral profile of one segment from each flow to be the representative spectral profile F_R for the entire flow. Figure 11 plots F_R for each traffic type per card. Once we selected the representative profile, we measured how well the spectral profiles of the other segments of the flow matched F_R. Table 1 summarizes the results.

To finish our analysis, we compare spectral profiles F_R between different flows and different cards. Reviewing the individual PSD plots (not shown due to space constraints) and spectral profiles in Figure 11, we draw the following conclusions.

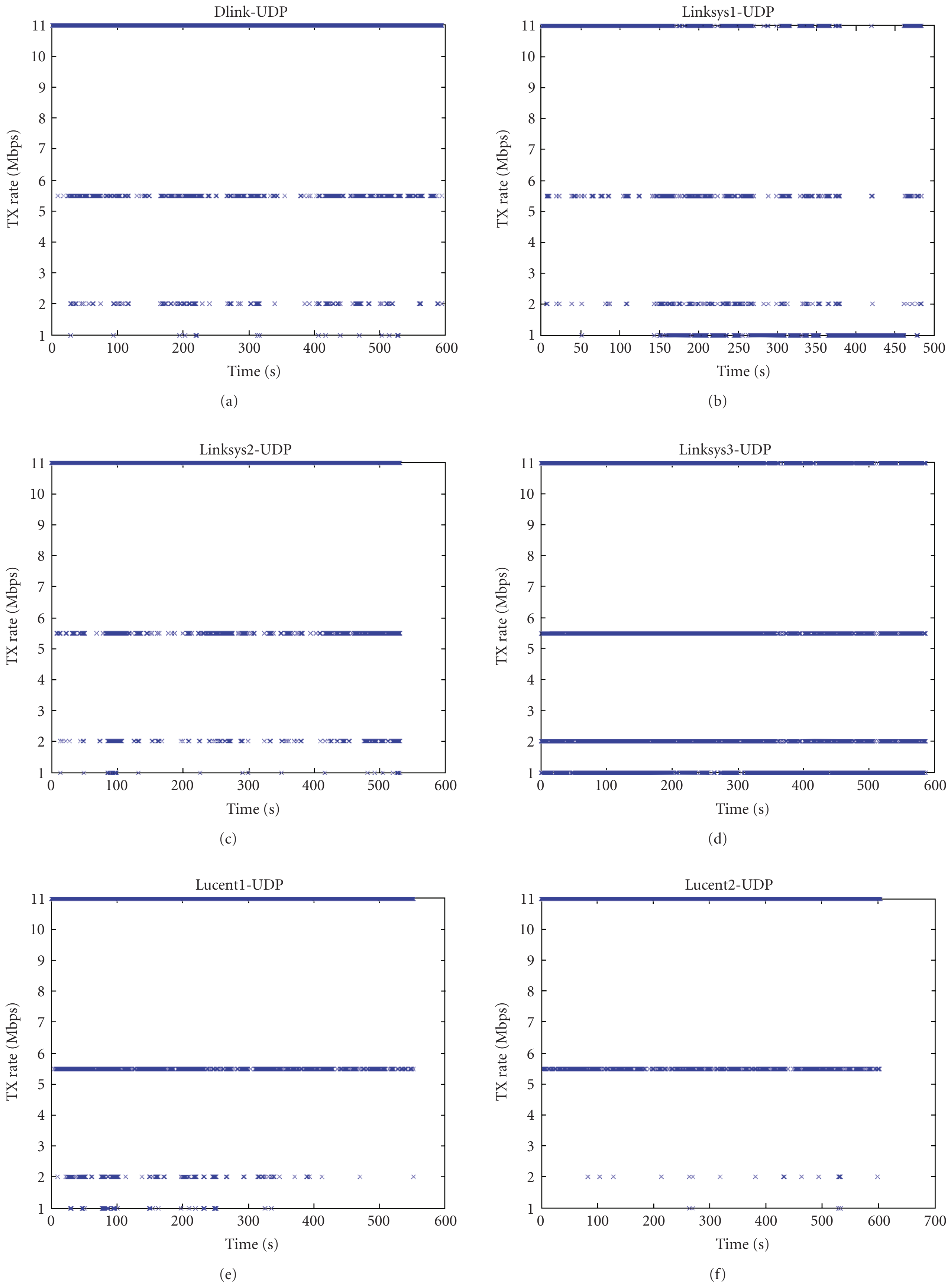
(i) For each card, the spectral profiles for the TCP flows overlap with the spectral profile for the UDP flow. This indicates that the spectral profile for rate switching is not sensitive to the type of traffic. Reviewing the complete spectra shows that there are some differences in the PSD between the different traffic types, but this occurs at frequency points where the magnitude of power is not as great as the power at the frequencies that they have in common.

(ii) For the Lucent card set, both of the cards (Lucent1 and Lucent2) had a similar spectral profile indicating that both of these cards implement the same rate-switching algorithm.

(iii) Within the Linksys card set, Linksys3 had a different spectral profile from the other two cards even though they all used the same software driver. We speculate that Linksys3 is a different version, which may have different hardware and/or firmware from the other two Linksys cards. This suggests that the manufacturer of the Linksys cards implemented the rate-switching algorithm within the hardware and/or firmware of the NIC, rather than the driver software.

(iv) The Dlink card behaved similarly to Linksys1 and Linksys2. Some versions of the DLink 650 card are based on the Intersil Prism chipset, which is the same chipset family used in Linksys cards. Dlink, Linksys1, and Linksys2 may all have had the same hardware. This would explain the similarities and coincide with the speculation we made above that the rate-switching algorithm for these types of cards is implemented in the hardware rather than the driver software.

Based on the observations we made above, we can classify the six NICs into three classes as shown in Table 2. Class 1 has a concentration of power primarily between 160 and 180 Hz. This indicates that data frames were most frequently sent between 5.56-millisecond and 6.25-millisecond intervals during rate switching. The class 2 card had a concentration of power around 70 Hz and 100 Hz (140 Hz, 200 Hz, and 300 Hz are considered as harmonics). Accordingly, class 2 sent data frames at 10-millisecond and 14-millisecond intervals during rate switching. The cards in class 3 primarily operate between 213 and 249 Hz, which correspond to data frames being sent every 4–4.7 milliseconds during rate switching.



(a) (b) (c) (d) (e) (f)

Figure 8: NICs at hotspot invoking rate switching during UDP session.

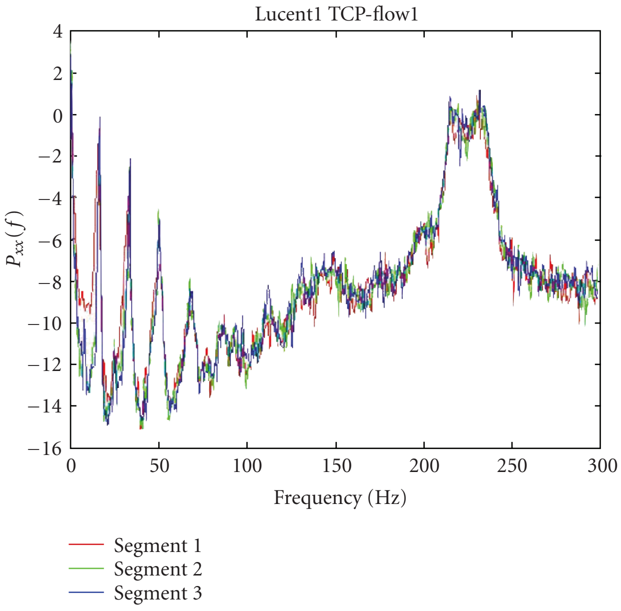


FIGURE 9: PSD of TCP-flow1 for the Lucent1 card.

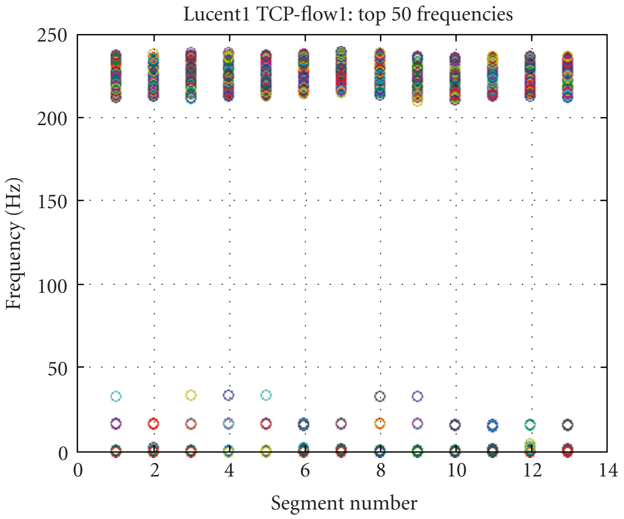


FIGURE 10: Plot of the top 50 frequency points that constitute the spectral profile for each segment of the Lucent1 TCP-flow1.

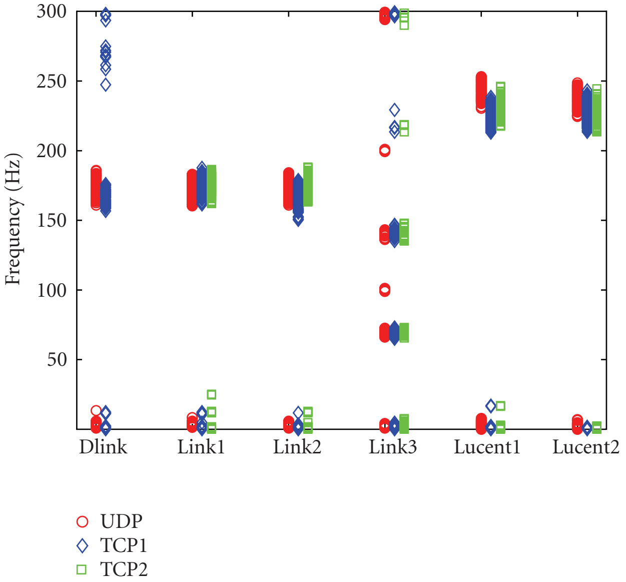


FIGURE 11: Plot of the spectral profile F_R for each NIC and traffic type.

7. CONCLUSION AND FUTURE WORK

Rate switching proved to be a stable attribute for identifying NICs. Additionally, we were able to make a distinction between cards manufactured by the same vendor. We also showed that rate switching behaved similarly for UDP and TCP traffic. Finally, we identified three classes of NICs based on the spectral characteristics of their rate-switching algorithm.

As we extend our work, it will be important to continue to explore the stability of the spectral profile. We have already considered different traffic types. It would also be useful to determine the impact of the composition of the host on the spectral profile. During our experiments, we used a single host. Differences in the composition of a host such as CPU speed, type of operating system, and host load may affect the spectral profile. If the spectral profile is sensitive to the composition of the host, we can increase the uniqueness of classification of wireless system even further than just the type of NIC.

We also plan to investigate other attributes, such as the setting of the user configurable parameters (i.e., RTS threshold, maximum retries, etc.), from which we can extract a spectral profile. To do so, we will need to consider other signal representations for wireless traffic. The current work primarily focuses on the arrival rate of data frames. We plan to examine other properties of a wireless frame to be encoded as a signal. For example, when investigating the impact of the setting of the RTS threshold, we could encode the arrival rate or interpacket delay of retransmitted frames. Additionally, we could weight the encoding process using the size of the retransmitted frame as well.

The current approach for deriving a spectral profile worked well for capturing the periodicities during rate switching. However, there may be other spectral features that are distinctive but unnoticed in the current approach, because they would be overshadowed by other features that exhibited a higher magnitude of power. An alternative may be to group adjacent frequencies as one feature. Another alternative may be to establish thresholds relative to the total power. A more robust technique for comparing profiles would be needed as the database of spectral profiles grows.

REFERENCES

[1] Cisco Security Agent, http://www.cisco.com/.
[2] ISS Proventia Desktop, http://www.iss.net/.
[3] Symantec Critical system Protection, http://www.symantec.com/.
[4] McAfee Entercept, http://www.mcafee.com/.

[5] Checkpoint Integrity, http://www.checkpoint.com/.

[6] Sana Primary Response, http://www.sanasecurity.com/.

[7] J. Wright, "Detecting Wireless LAN MAC Address Spoofing," www.net-security.org/dl/articles/wlan-mac-spoof.pdf.

[8] ReefEdge, http://www.tribecaexpress.com/reefedge.htm.

[9] AirDefense, http://www.airdefense.net/.

[10] AirMagnet, http://www.airmagnet.com/.

[11] WiMetrics, http://www.wimetrics.com/.

[12] iPass, http://www.ipass.com/services/services_deviceid.html/.

[13] Cellular Companies Fight Fraud, http://www.decodesystems.com/mt/97dec/.

[14] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase," in *Proceedings of the Internet and Information Technology (CIIT '04)*, St. Thomas, US Virgin Islands, November 2004.

[15] T. Kohno, A. Broido, and K. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[16] C. L. Corbett, R. A. Beyah, and J. A. Copeland, "A passive approach to wireless NIC identification," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, Istanbul, Turkey, June 2006.

[17] Y. Fyodor, "Remote OS detection via TCP/IP stack fingerprinting," October 1998, http://www.insecure.org/nmap/nmap-fingerprinting-article.txt.

[18] O. Arkin and F. Yarochkin, "Xprove v2.0: A Fuzzy Approach to Remote Active Operating System Fingerprinting," http://www.sys-security.com/archive/papers/Xprobe2.pdf, 2002.

[19] "Agere's WiFi chipset reaches 150Mbit/s," http://www.electronicsweekly.com/Article5144.html.

[20] "IEEE 802.11 specification," http://standards.ieee.org/getieee802/802.11.html.

[21] M. Lacaga, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM/MSWIM '04)*, Venice, Italy, October 2004.

[22] P. Chevillat, J. Jelitto, A. Noll Barreto, and H. L. Truong, "A dynamic link adaptation algorithm for IEEE 802.11a wireless LANs," in *Proceedings of the IEEE International Conference on Communications (ICC '03)*, vol. 2, pp. 1141–1145, Anchorage, Alaska, USA, May 2003.

[23] GigaMobile, "Application-directed automatic 802.11 rate control," December, 2005 https://doc.freeband.nl/dscgi/ds.py/Get/File-28445/GigaMobile-D3.16.pdf.

[24] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[25] S. Choi, K. Park, and C.-K. Kim, "On the performance characteristics of WLANs; revisted," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS '05)*, Banff, Alberta, Canada, June 2005.

[26] C.-M. Cheng, H. T. Kung, and K.-S. Tan, "Use of spectral analysis in defense against DoS attacks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 3, pp. 2143–2148, Taipei, Taiwan, November 2002.

[27] A. Hussain, J. Heidemann, and C. Papadopoulos, "Identification of repeated attacks using network traffic forensics," Tech. Rep. ISI-TR-2003-577b, USC/Information Sciences Institute, Marina del Rey, Calif, USA, August 2003.

[28] C. Partridge, D. Cousins, A. W. Jackson, R. Krishnan, T. Saxena, and W. Timothy Strayer, "Using signal processing to analyze wireless data traffic," in *Proceedings of the Workshop on Wireless Security (WiSe '02)*, pp. 67–76, Atlanta, Ga, USA, September 2002.

[29] J. McClellan, R. Schafer, and M. Yoder, *Signal Processing First*, Prentice Hall, Englewood Cliffs, NJ, USA, 2003.

[30] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[31] Signal Processing Toolbox, http://www.mathworks.com/access/helpdesk/help/toolbox/signal/.

[32] The linux-wlan^{TM} Project, http://www.linux-wlan.org/.

[33] Tcpdump, http://www.tcpdump.org/.