IJIS (2008) 7:335–348
DOI 10.1007/s10207-007-0053-7

REGULAR CONTRIBUTION

Passive classification of wireless NICs during active scanning

Cherita L. Corbett · Raheem A. Beyah · John A. Copeland

Received: 6 July 2006 / Accepted: 15 December 2007 / Published online: 8 January 2008

Abstract 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 present a statistical analysis and 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. We focus on active scanning, 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 that are observable through traffic analysis. Our mechanism for NIC identification uses signal processing to analyze the periodicity embedded in the wireless traffic caused by active scanning. 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, with zero false positives, using the spectral profile. Finally, we infer where, in the NIC, the active scanning algorithm is implemented.

Keywords Network security · Wireless security · Host identification

C. L. Corbett
Computer & Network Security Group,
Sandia National Laboratories, 7011 East Avenue, MS 9011,
Livermore, CA 94550, USA
e-mail: clcorbe@sandia.gov

R. A. Beyah (✉)
Department of Computer Science, Georgia State University,
34 Peachtree Street, Suite 1451, Atlanta, GA 30303, USA
e-mail: rbeyah@cs.gsu.edu

J. A. Copeland
Communications Systems Center, School of Electrical
and Computer Engineering, Georgia Institute of Technology,
75 Fifth Street, Atlanta, GA 30308, USA
e-mail: jcopeland@ece.gatech.edu

1 Introduction

Computer networks have become more ubiquitous. Further, 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 only problem of 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 access to 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 are 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 active scanning, 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. We motivate the need for and apply signal processing to analyze the periodicity embedded in the wireless traffic caused by active scanning. A spectral profile is created from the periodic components of the traffic and used for the identity of the wireless NIC.

The remaining of this paper is organized as follows. Section 2 discusses related work. In Sect. 3 we discuss the composition of a wireless NIC and present opportunities for distinguishing between different types of NICs. Sect. 4 presents the rational 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 Sect. 4. In Sect. 5, we present the experimental setup, statistical and spectral analysis techniques used to distinguish between NICs during active scanning, and qualitative and quantitative results. Section 6 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 medium access control (MAC) address spoofing. Many of the attack tools (e.g., FakeAP, AirJack, and Wellenreiter) aimed at obtaining unauthorized access rely on spoofing 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 non-fragmented 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 does not exhibit alarming behavior as he had the proper credentials.

An alternative method for defending a wireless network from unauthorized access is to establish an identity for legitimate systems. Normally the MAC address of the 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 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 for 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 it 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 of including 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 used spectral analysis to distinguish between network cards manufactured by different vendors. In particular, we focused on the rate-switching algorithm that is vaguely specified in the 802.11 specification. An empirical analysis was conducted at a local hotspot to characterize rate switching. The results of this work were a unique PSD for each card that can be used as a spectral fingerprint. Although this approach proved effective, it should be noted that it assumes some level of RF interference. Even though this is highly likely the majority of the time [16], there are still instances where there will be minimal interference and a node will not trigger rate switching.

3 Wireless network interface card

A wireless 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 semi-permanently embedded into ROM to control the hardware. It works to communicate between the hardware and the driver software. Driver software accepts generic I/O commands from the OS of the host and then converts them into instructions 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, etc. 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 inter-arrival 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 scanning mechanism to distinguish between NICs.

3.1.1 Scanning

Scanning is one of the primary 802.11 MAC functions, whereby a client seeks to discover available wireless networks to join. The standard [20] defines both passive and active scanning. In passive scanning mode, the NIC listens on individual channels for beacon frames from access points (APs), noting the corresponding signal strengths and other information about the access points. The NIC uses this information to decide which access point to use. Active scanning involves the broadcasting of probe request frames and the subsequent processing of received probe response frames. Active scanning enables a NIC to solicit an immediate response from an AP, without waiting on beacon transmissions. Active scanning is the default mode for NICs. For our research, we solely focus on active scanning, hereafter referred to as scanning.

The scanning mechanism is most often implemented in the device driver of the NIC. It is engaged when an NIC is first turned on as well as during a handoff procedure. A handoff is the change of AP to which a station is connected and occurs when the connection with the present AP degrades below a threshold. The guidelines set by the IEEE 802.11 MAC standard for the scanning procedure are as follows (modified for brevity):

For each channel to be scanned:

1. Wait until *ProbeDelay* time has expired.
2. Send a probe request with broadcast destination, SSID and broadcast BSSID.
3. Start a *ProbeTimer*.
4. If medium idle (i.e., there is neither a response nor any kind of traffic) when *ProbeTimer* reaches *MinChannelTime*, scan the next channel; else, when *ProbeTime* reaches *MaxChannelTime*, process all received probe responses and scan next channel.

ProbeDelay is the delay to be used prior to transmitting a probe request frame on a new channel. *MinChannelTime* is the minimum amount of time to spend on each channel. *MaxChannelTime* is the maximum amount of time to spend on each channel.

Recent research [21,22] has analyzed the scanning process for its impact on the handoff performance. Contributions have been made in developing new schemes that minimize the amount of time a NIC spends scanning. The authors noted that during their hand-off measurements, the scanning duration varied among cards by different vendors. We exploit variations in the scanning mechanism to identify cards by different vendors.

The scanning procedure is vaguely specified, consequently forcing vendors to implement proprietary solutions. The scanning process has several parameters that can vary per vendor including:

- values for *ProbeDelay*, *MinChannelTime*, *MaxChannelTime*
- number of probe request frames to transmit per channel
- delay between probe request frames on the same channel
- channel probe frequency
- order of channels to probe

The performance of the scanning mechanism depends upon the setting of these parameters and defines the behavior of the wireless stream. We apply spectral analysis on the traffic stream to extract the traffic patterns imposed by the scanning mechanism to identify NICs by different vendors.

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 content.

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 vs. 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 inter-relationship 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 [23] applied spectral analysis to distinguish between normal TCP traffic and Denial-of-Service (DoS) attack traffic in aggregated, high-volume traffic. Hussain et al. [24] 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) of the attack host changed. This improved classification of the attacker beyond the type of attack tool. Partridge et al. [25] 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 active scanning mechanism.

4.2 Signal representation

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, etc.

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 $(t, t+T]$.

$$x(t) = x(nT), \text{ where } n = 0, 1, 2, 3, \ldots$$

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 [26] 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 [27], 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 the 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 inter-relationship within a signal. It does so by using the discrete-time Fourier transform (DFT) of the samples of a signal and taking the magnitude squared of the result. The PSD P_{xx}, of a signal of length L is given in (1):

$$\hat{P}_{xx}(f) = \frac{|X_L(f)|^2}{f_s L} \tag{1}$$

where the discrete Fourier transform, X_L is given in (2):

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

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 [28] is implemented as follows:

1. The input signal vector x is divided into k overlapping segments according to segment length l and number of overlapping samples *noverlap*.
2. The specified windowing function w is applied to each segment of x.
3. An *nfft*-point FFT is applied to the windowed data.
4. The modified periodogram of each windowed segment is computed.

5. The set of modified periodograms is averaged to form the spectrum estimate $\hat{P}_{xx}(f)$.
6. 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.

The number of segments k that x is divided into is calculated as:

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

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 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 Fig. 1. The algorithm above locates N frequency points that exhibit the greatest amount of power to constitute a spectral profile $F = \{f_1, f_2, f_3, \ldots f_N\}$. These key frequency points estimate the most prevalent arrival rates of frames in a wireless stream.

5 NIC IDENTIFICATION USING SCANNING

The traffic generated during the scanning process presents itself as an opportunity for distinguishing between wireless NICs manufactured by different vendors. We apply signal processing to capture the temporal behavior of the scanning process. In the following sections we explain the experimental setup, apply our spectral analysis technique, and evaluate the results.

5.1 Experimental setup

For our experiments we used one access point, one client station to engage the scanning procedure, and three wireless sniffers collected traffic for analysis. Figure 2 illustrates the layout. All experiments were done in an isolated environment where there was little or no other wireless activity from neighboring networks.

5.1.1 Client setup

We used a 1 GHz Celeron Toshiba laptop with Linux Redhat 9 as the wireless client. Wireless cards interface with the laptop via the PCMCIA slot. We tested six different cards: two Lucent/Orinoco Gold cards, two Linksys WPC11 cards, a D-Link DWL-650 card, and a Cisco 350 card. During the experiments, the Lucent cards used the *orinoco_cs* software driver. The Linksys and DLink cards used the *prism2_cs* software driver. The Cisco card used the *airo_cs* software driver. The experiments were conducted in the following manner for each card. A card was inserted into the PCMCIA slot and ejected after 4 min completed. The scanning process is automatically engaged upon inserting the card. We repeated this process 100 times for each card. To automate this process we wrote a perl script to execute the *cardctl* command to turn the PCMCIA slot on (this denoted the insertion of the NIC) and off (this denoted the ejection of the NIC). To ensure synchronization with the traffic collection process, we used the Network Time Protocol (NTP) through a wired Ethernet connection and used the *crontab* command to schedule execution of our perl script periodically.

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

Fig. 1 Algorithm for comparing spectra

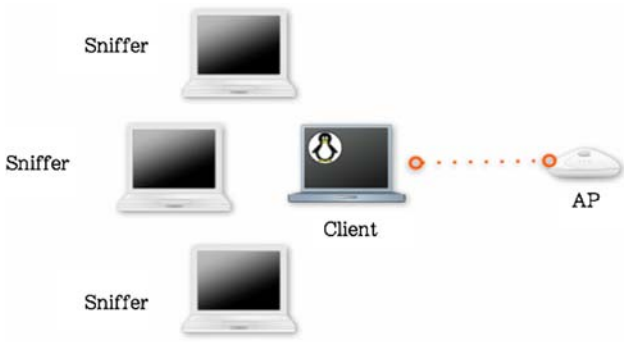

Fig. 2 Experimental setup

5.1.2 Data collection

During the scanning process, the client broadcast probe request frames on different channels. We used three 3 GHz Pentium 4 Toshiba laptops with Linux Redhat 9 as sniffers to collect traffic on multiple channels independently. Each sniffer was configured with an internal Atheros wireless NIC and an external wireless NIC inserted through the PCMCIA slot. Two Linksys WPC11 cards were used as the external NICs. As a result, the available hardware limited us to collecting traffic on 5 channels (1 through 5) simultaneously while the client was scanning.

To be able to see the raw IEEE 802.11 frames on a particular channel, each card was put into monitor mode on an assigned channel using the *iwconfig* and *wlanctl-ng* utilities [29]. The sniffers used *tcpdump* [30] to collect the frames with timestamps into a traffic capture file. To automate the data collection process each sniffer used a perl script to execute the *tcpdump* command. To ensure that the capture covered the scanning period, *crontab* was used to schedule execution of the perl script on the sniffers at the same time as the execution of the perl script at the client. Synchronization was maintained with the client using the NTP through a wired Ethernet connection. The data collection process resulted in 100 traffic capture files for each channel per card (total of 3000 traffic capture files).

5.2 Statistical analysis

Before applying spectral analysis, we conducted a statistical evaluation on the scanning traffic generated. For each card, an evaluation was done per channel. First, we counted the number of probe request frames sent during each experimental trial for each channel per card. Figure 3 shows the results and Table 1 summarizes these results. The results show that the Cisco and Lucent cards were much more aggressive in sending probes than the other cards. For all of the cards, channel 4 was probed more often than channels 1, 2, 3, and 5.

Next we measured the length of time each card probed a channel within the 4 minute observation period. The results

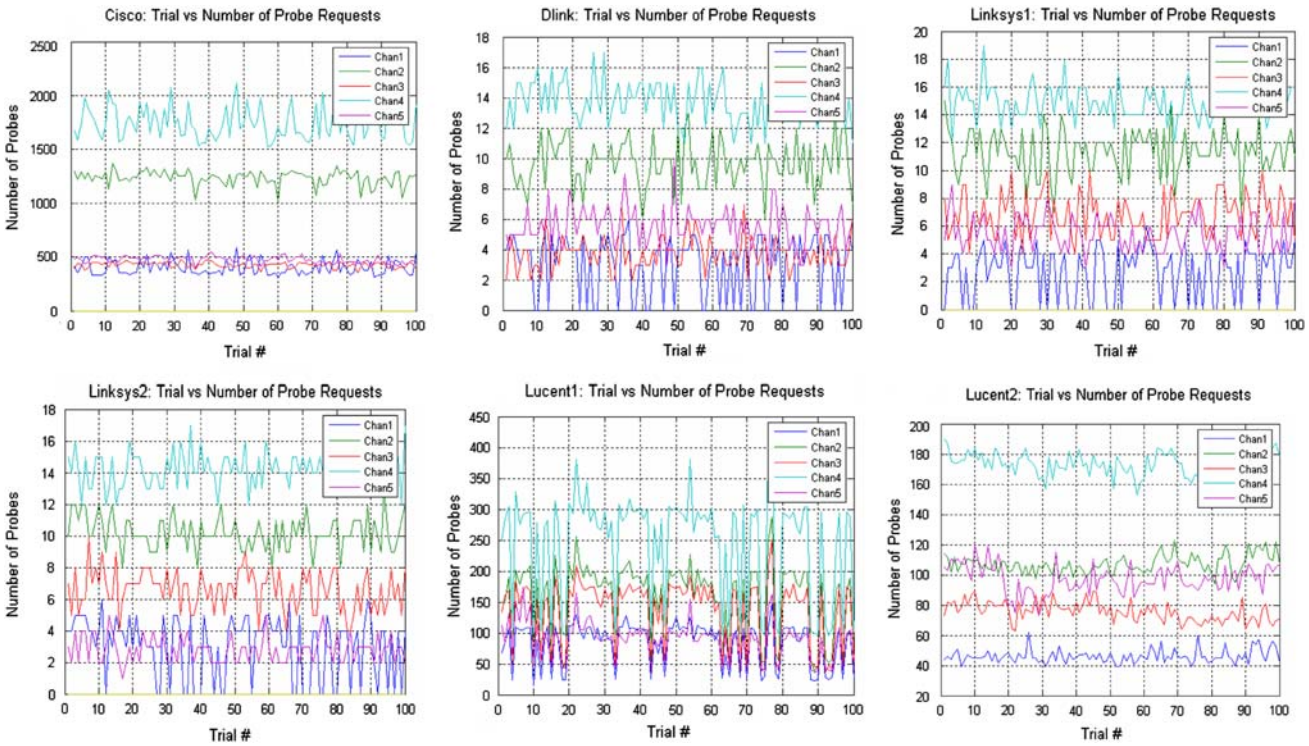

Fig. 3 Number of probe request frames sent by each wireless NIC

Table 1 Number of probe request frames transmitted

		Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Cisco	min	313	1021	358	1523	395
	max	585	1374	477	2141	546
	mean	398.86	1235.16	426.43	1745.65	475.99
	median	376	1247	430	1721	474.5
Dlink	min	0	6	2	11	3
	max	6	13	7	17	10
	mean	3.02	9.7	3.77	13.79	5.6
	median	4	10	4	14	6
Linksys 1	min	0	7	4	12	3
	max	6	15	10	19	9
	mean	2.7	11.34	6.86	15.03	5.44
	median	3	11	7	15	5
Linksys 2	min	0	8	4	12	1
	max	6	13	10	17	5
	mean	2.98	10.23	6.63	14.29	2.85
	median	4	10	7	14	3
Lucent 1	min	15	37	36	93	36
	max	150	288	251	439	175
	mean	83.09	155.24	136.18	245.14	88.85
	median	103.5	181.5	158	284.5	94.5
Lucent 2	min	39	94	63	153	73
	max	62	123	89	191	120
	mean	46.1	107.29	75.66	174.4	97.25
	median	45	107	75	174	96.5

are shown in Fig. 4 and summarized in Table 2. The Cisco card scanned all the channels for almost the entire 4 min. The Lucent cards scanned most of the channels for almost the entire 4 min. However, the Dlink and Linksys cards scanned most of the channels for about 2 min.

We also examined the manner in which the cards probed a channel. We observed that for most channels, the Cisco card would send a burst of probes almost every second. This can be seen by the stair–step slope for the Cisco card in Fig. 5. The Lucent cards also exhibited a similar behavior, sending a burst of probes, then waiting a period of time (typically 2, 8, or 10 s) to transmit the next burst of probes (Fig. 5). The Dlink card sent a burst of probes and waited about 115 s to transmit one more probe request frame (Fig. 5). The Linksys cards behaved in the same manner as the Dlink card as shown in Fig. 5. While our statistical analysis revealed some discerning properties among different cards, we cannot solely rely on this information because it is most accurate when there is a complete capture of the scanning process, which may not always be available. Signal processing examines the inter-relationship between a sequence of events, which allows the temporal properties to be extracted without requiring the complete view of the scanning process. We discuss the use of spectral analysis in the next section.

5.3 Spectral analysis

We processed the traffic collected in the capture files using the signal processing technique discussed in Sect. 4. The probe request frames and associated time stamps were extracted from the traffic capture file to represent the time series of events. Next, we sampled the time series at a rate of 500 Hz, counting the number of probe request frames that arrive in each 0.002 s bin. The next step was to apply the Welch method to estimate the power spectral density. We used a 1024-point FFT, a segment length of 64 data points, an overlap of segments by 32 data points, and the Hanning window to configure the Welch method. With a sampling rate of 0.002 s, we were able to observe periodicity in the scanning process over a range of 250 Hz. Processing the capture files in this manner generated 100 PSDs for each channel per card. Figure 6 illustrates a subset of the output graphs for each of the cards on channel 4. To keep the graphs legible, we avoid plotting

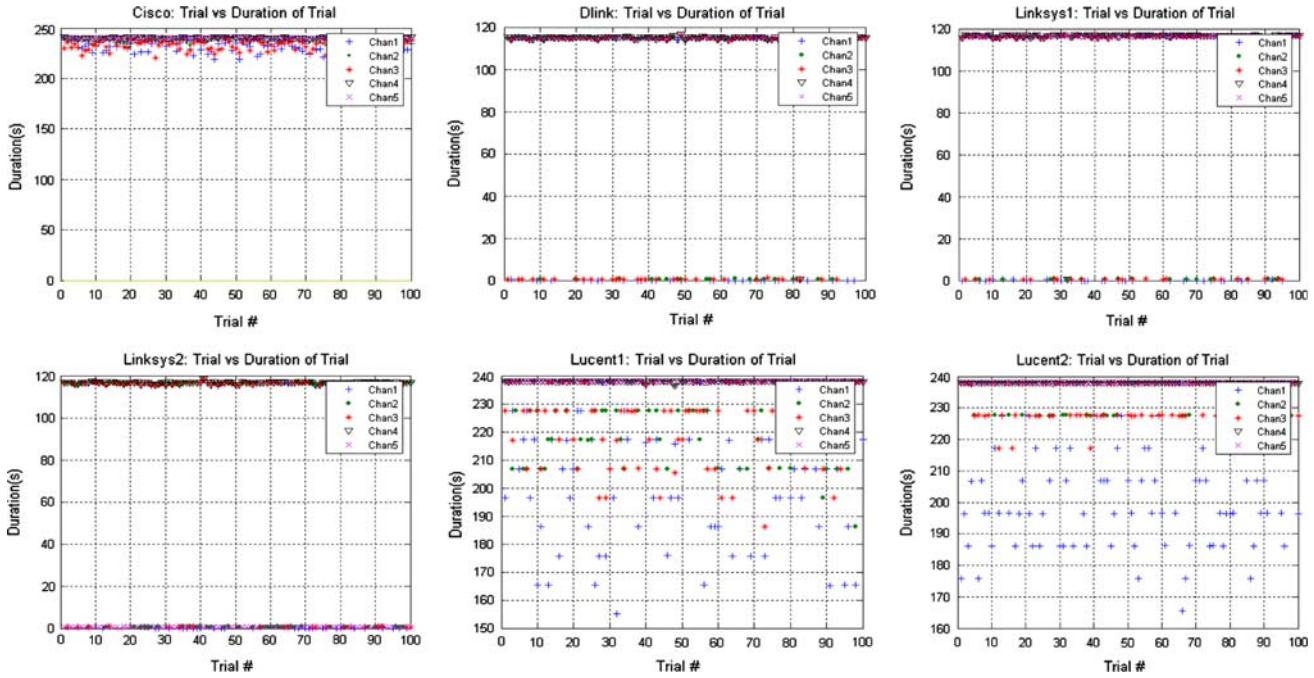

Fig. 4 Duration of probing for each wireless NIC

Table 2 Duration of probing (seconds)

		Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Cisco	min	219.09	231.41	219.84	238.17	236.15
	max	240	240	240	240	240
	mean	232.0829	237.502	232.781	239.8895	239.7506
	median	232.7	237.895	233.66	240	240
Dlink	min	0	0.46614	0.34602	0.70156	0.5176
	max	115.36	115.44	116.45	116.6	116.4
	mean	75.80909	94.29148	78.29326	113.8627	113.6692
	median	114.705	114.94	114.805	115.125	114.92
Linksys 1	min	0	0.57264	0.4843	0.68081	115.37
	max	116.93	116.91	116.91	117.08	116.93
	mean	74.55683	94.44631	89.81147	115.4179	116.4071
	median	116.395	116.57	116.51	116.74	116.58
Linksys 2	min	0	0.50257	0.45159	115.58	0
	max	118.13	118.21	118.28	118.33	0.79882
	mean	78.00183	71.29663	92.14231	116.5573	0.53967
	median	116.325	115.745	116.565	116.74	0.53106
Lucent 1	min	155.04	186.1	186.12	236.52	236.51
	max	237.92	237.97	237.97	238	238.05
	mean	209.4641	226.6719	225.0143	237.8896	237.8829
	median	217.075	227.57	227.55	237.92	237.91
Lucent 2	min	165.42	227.47	217.22	237.83	237.76
	max	238.02	238.09	238.05	238.1	238.09
	mean	206.8664	235.3699	232.9854	237.9971	237.9744
	median	206.85	237.94	237.92	238	237.98

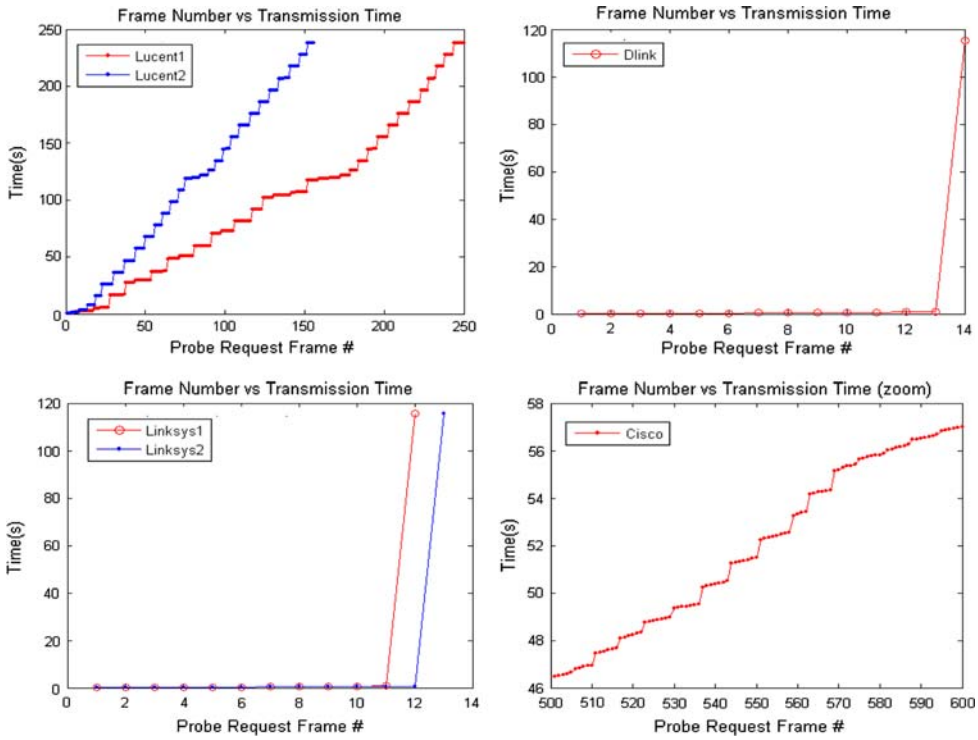

Fig. 5 Example of burstiness in the scanning mechanism for each NIC

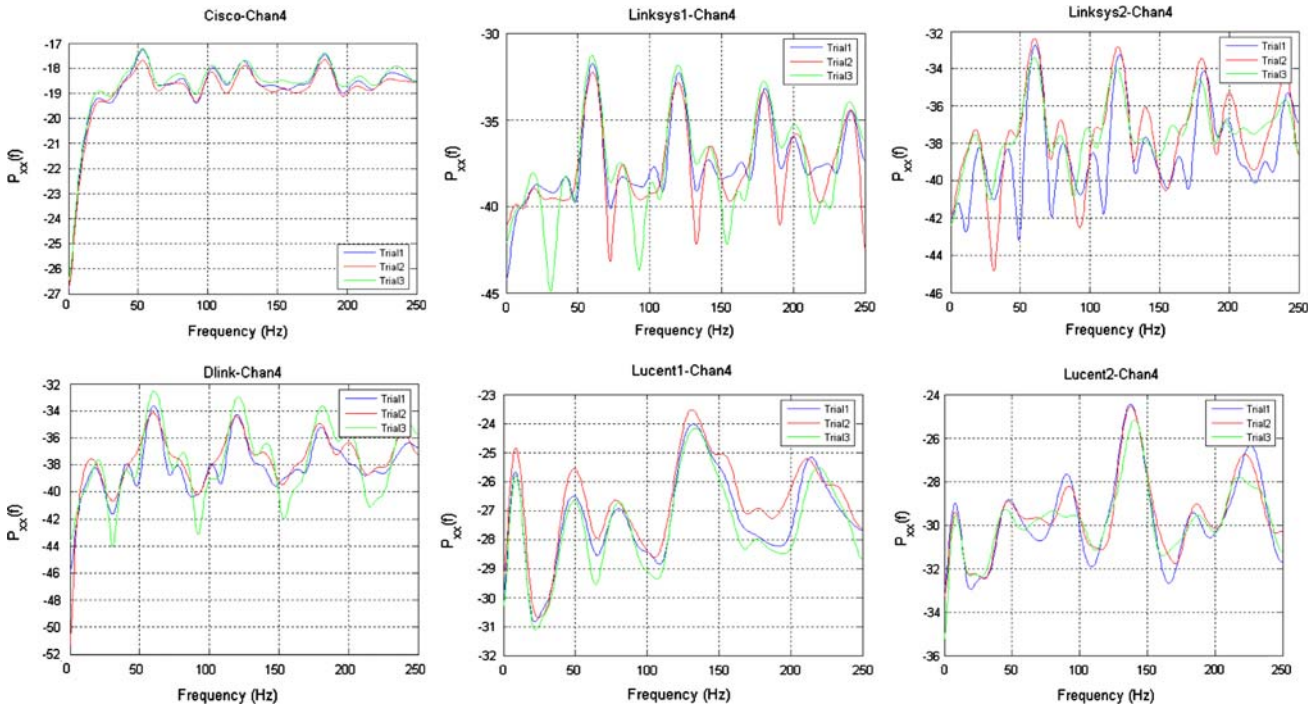

Fig. 6 PSDs of scanning process on channel 4 for all NICs

every PSD and only selected three PSDs to plot per graph. We chose channel 4 because it was favored by all of the scanning algorithms and had the most stable spectral profile for each card.

5.3.1 Qualitative results

The peaks in the PSD estimates show periodicity in the scanning process on all channels for all cards. The frequency

points at which the peaks occur correspond to the transmission rates of the probe request frames. The magnitude of the peaks is proportional to how often the transmission rate occurs. General observations can be made by visually comparing the PSD graphs for different channels and different cards. In the following, we discuss our observations.

Cisco

- The PSD estimates for repeated trials on the same channel are similar.
- The PSD estimates for channel 1 has the most prominent peaks around 50, 100, and 185 Hz.
- The PSD estimates for channels 2, 3, and 4 have the most prominent peaks around 55, 110, 125, and 185 Hz.
- The PSD estimates for channel 5 has the most prominent peaks around 50 and 185 Hz.

Dlink

- The PSD estimates for repeated trials on the same channel are similar.
- The choice in the sampling rate resulted in the occurrence of harmonics.
- The fundamental frequency for scanning on channel 1 is 20 Hz.
- Channels 3 and 5 exhibit the same behavior as channel 1.
- The fundamental frequencies for scanning on channel 2 are 20, 40, and 60 Hz. The prominent peak at 60 Hz contains significantly more power, indicating that probes are sent at this rate much more regularly than the other frequencies.
- Channels 4 and 6 exhibit the same behavior as channel 2.

Linksys

- The PSD estimates for repeated trials on the same channel are similar.
- The PSD estimates are similar between both of the Linksys cards on all channels.
- The PSD estimates are similar to Dlink for channels 1, 2, 4, and 5 (see observations listed for Dlink card).
- The fundamental frequencies for scanning on channel 3 are 20, 40, and 60 Hz. However, magnitude of difference at these frequency points is small. In some trials, the PSD for channel 3 is similar to channel 1.

Lucent

- The PSD estimates for repeated trials on the same channel are similar.
- The PSD estimates between both of the Lucent cards are similar for channels 4, 5, and 6.
- The PSD estimates for channels 4, 5, and 6 have the most prominent peaks around 135 and 215 Hz.

5.3.2 Quantitative results

The previous section examined the PSD estimates visually, which allowed comparison of the complete spectra. To numerically compare spectra we used the approach discussed in Sect. 4.4, which uses a subset of the PSD values. For our analysis of the scanning process we elected to use 50 frequency points from the PSD that exhibited the greatest amount of power. These key frequency points represent the most prevalent sending rates of the probe request frames. For each trial, we let the set of frequencies $F = \{f_1, f_2, f_3, \ldots f_{50}\}$ with the greatest amount of power constitute the spectral profile that we use to compare spectra. Figure 7 plots the set F for each trial on channel 4 for all cards. The formation of a horizontal line at a particular frequency range is indicative of the similarity in the spectral content among trials on the same channel. For example, Fig. 7 shows that the spectral profile for all trials on channel 4 of Lucent2 contain frequencies between 130–150 and 218–225 Hz. Conversely, if the data points are spurious, then the spectral content among the trials on the same channel are different at those frequencies. Even though we are only working with a subset of the values given by the PSD, the results plotted in these graphs reiterate the observations made above in the qualitative results section.

For each channel we arbitrarily selected 1 out of the 100 trials to use as the representative trial $T_{\mathbf{R}}$ and its corresponding profile to be the representative spectral profile $F_{\mathbf{R}} = \{f_1, f_2, f_3, \ldots f_{50}\}$ for that channel. Next, we compared the spectral profile of the remaining trials to $F_{\mathbf{R}}$ to measure robustness of $F_{\mathbf{R}}$. The results of the comparisons on channel 4 are shown in Tables 3, 4, 5, 6, 7 and 8. The first column of the table groups the contiguous frequencies of $F_{\mathbf{R}}$ into frequency ranges. The second column tells what portion of $F_{\mathbf{R}}$ is within a particular range. The last column is the percent of trials with a spectral profile that contain frequencies in the same range as the frequencies as $F_{\mathbf{R}}$. The last row is the most important and indicates the percent of trials with a spectral profile containing frequencies within all the ranges associated with $F_{\mathbf{R}}$. Our results showed that the comparison of spectral profiles matched best on channel 4, where 90% or more of the trials matched the spectral profile $F_{\mathbf{R}}$ of the representative trial. None of the cards were misclassified, resulting in zero false positives and a false negative rate ranging between 3 and 10% (Tables 3,4, 5, 6, 7, 8).

Next, we used the spectral profile $F_{\mathbf{R}}$ generated for channel 4 to compare NICs. Figure 8 plots the frequency values of the spectral profile of channel 4 for each NIC. Cards of the same type had a similar spectral profile. Additionally, the Dlink card had a similar profile to the Linksys cards. This can be attributed to the fact that Dlink and the Linksys cards used the same *prism2_cs* driver software, which

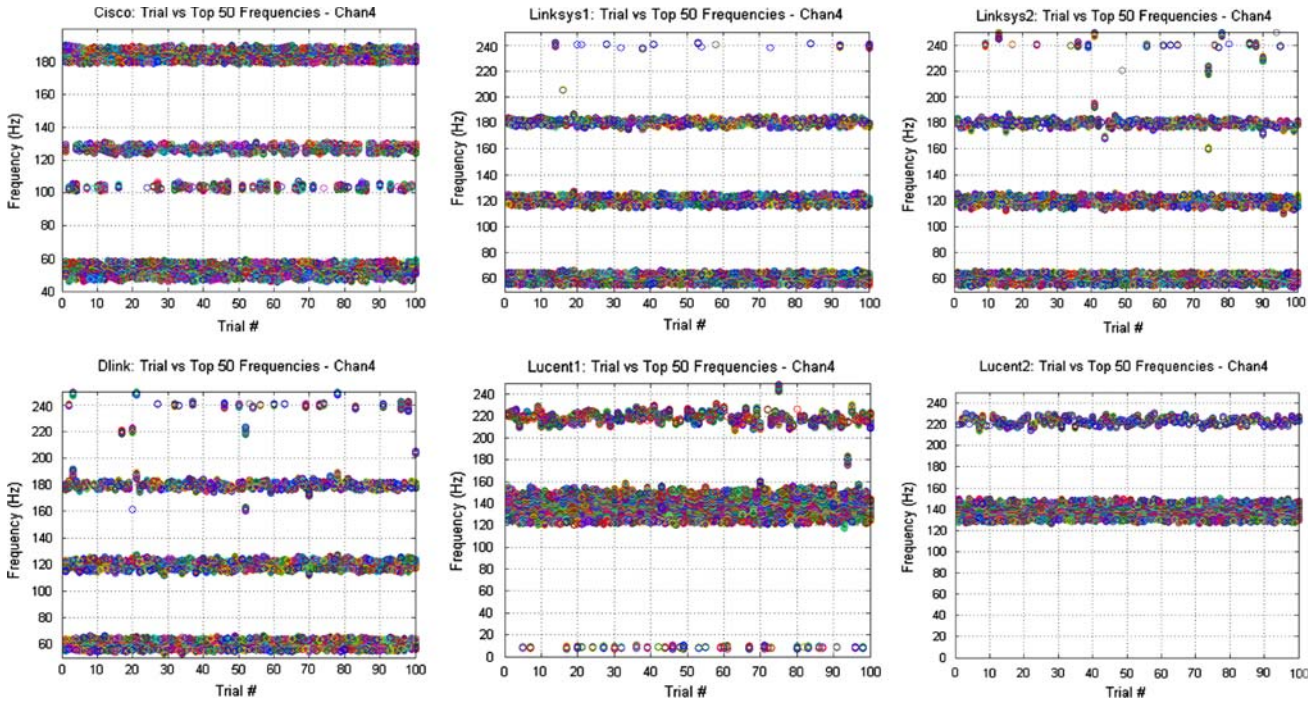

Fig. 7 Plot of the top 50 frequencies for all trials on channel 4 of the Cisco, Dlink, Linksys, and Lucent cards

Table 3 Ciisco Channel 4

Frequency Range (Hz)	Percent of F_R	Percent Match
48.828–58.594	39%	100%
125–129.39	19%	90%
179.69–190.43	43%	100%
All		90%

Table 4 Dlink Channel 4

Frequency Range (Hz)	Percent of F_R	Percent Match
55.176–64.453	40%	100%
115.72–123.54	34%	100%
176.27–182.13	26%	90%
All		90%

Table 5 Linksys 1 Channel 4

Frequency Range (Hz)	Percent of F_R	Percent Match
55.664–64.941	40%	100%
116.7–124.51	34%	100%
177.73–183.59	26%	97%
All		97%

Table 6 Linksy 2 Channel 4

Frequency Range (Hz)	Percent of F_R	Percent Match
55.176–64.453	40%	100%
115.23–123.54	36%	100%
176.76–182.13	24%	94%
All		94%

Table 7 Lucent1 Channel 4

Frequency Range (Hz)	Percent of F_R	Percent Match
126.46–146	82%	100%
221.19–225.1	18%	90%
All		90%

Table 8 Lucent2 Channel 4

Frequency Range (Hz)	Percent of F_R	Percent Match
131.84–149.41	74%	100%
218.75–224.61	26%	94%
All		94%

encouraged us to hypothesize that scanning is implemented primarily in the driver software. For Dlink and both of the Linksys cards, the concentration of $F_{\mathbf{R}}$ is between 55 and 65 Hz (frequency ranges 116–125 and 176–184 Hz are harmonics). This indicates that the transmission rate of probe request frames is most often sent around 0.0167 s. The $F_{\mathbf{R}}$ for both of the Lucent cards indicate that most of the power is concentrated between 126–149 Hz and between 210 and 225 Hz. This corresponds to a transmission rate between 0.0076 and 0.0079 s and a second transmission rate between

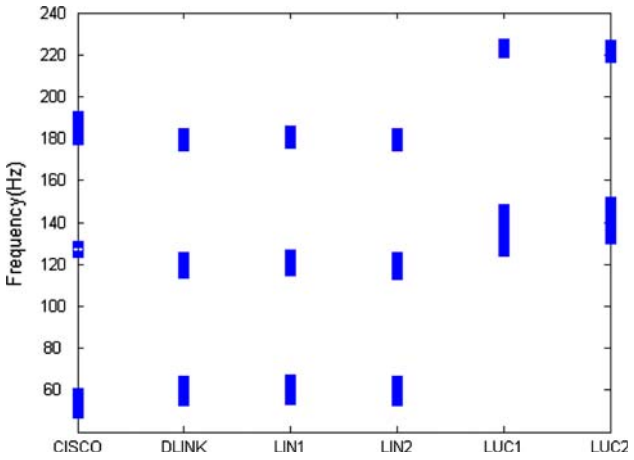

Fig. 8 Plot of spectral profile $F_{\mathbf{R}}$ for channel 4 of each card

0.0044 and 0.0048 s. The concentration of $F_{\mathbf{R}}$ for Cisco card lies in three ranges: 49–59, 125–129, and 180–190 Hz. Accordingly, probe request frames were sent at rates between 0.0169 and 0.0204 s, 0.0078 and 0.008 s, and 0.0053 and 0.0056 s.

5.4 Discussion

The scanning mechanism implemented in the NIC is a viable attribute to identify cards. A statistical analysis showed that the scanning mechanism of the Cisco card and Lucent cards was more aggressive than the other cards. The Cisco and Lucent cards transmitted more probes, sent probes at a faster rate, and probed channels for a longer period of time. Although it appears that statistical analysis can distinguish between cards, it cannot be used alone because statistics can be misleading if the entire scanning process is not captured. Spectral analysis proved to be a stable approach for analyzing the scanning process as evidenced by the ability to reproduce the PSD for repeated trials. Ideally one would like to listen on all channels at once to monitor the scanning process of a client to develop a scanning profile. However, sniffing on all channels is impractical. Additionally, profiling on certain channels was shown to be better than others, as scanning algorithms tend to favor some channels over others, because vendors try to anticipate the channels most likely to offer network connectivity to minimize the amount of time the card spends scanning. Out of the subset of channels we examined, our results showed that channel 4—(1) is the best channel for profiling wireless NICs; and (2) gives enough information to accurately profile a host. Results showed that there was more regularity and stability in the way the scanning mechanism of all card types probed channel 4. This is also evident by the fact that there were significantly more probes on channel 4 than channels 1, 2, 3, and 5. As a result, there was more communication traffic to observe and a better opportunity for identifying card types. Once we compared the spectral profile of all the cards for channel 4 we showed that different cards manufactured by the same vendor and used the same driver had the same spectral profile. We were also able to discern between Cisco, Lucent and Linksys/Dlink. Linksys and Dlink had identical spectral profiles likely because they used the same software driver. This led us to conclude that the scanning mechanism is implemented in the software driver of the NIC.

6 Conclusion and future work

We focused on differences in the implementation of the active scanning mechanism, a function required by the 802.11 standard, for establishing the identity of wireless NICs. We presented a statistical analysis of the active scanning process and used signal processing to analyze the periodicity embedded in the wireless traffic caused by active scanning. We developed a technique to create a spectral profile from the periodic components of the traffic to use as the identity of a wireless NIC. Using the spectral profile generated from the scanning mechanism we were able to discern between cards manufactured by different vendors. Although we only monitored five wireless channels as a result of hardware limitations, we showed that a single channel can be used to profile a wireless NIC. The proposed scheme had a zero false positive rate throughout the hundreds of trials and had a false negative rate ranging from 3 to 10%.

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 restrict the 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), 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 probe request frames and data frames. We plan to examine other properties of a wireless frame to encode as a signal. For example, when investigating the impact of the setting of the RTS threshold, we could encode the arrival rate or inter-packet 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 important active scanning. However,

there may be other spectral features that are distinctive but are unnoticed in the current approach because it 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.

One shortcoming of the proposed approach is that if the attacker knows the method of detection used, it is plausible that he or she may purchase a card manufactured by the same vendor. It is very expensive for a company to obtain proprietary hardware; however, as discussed in Sect. 5.4, the driver software is key to identifying the NIC and it is much easier to acquire proprietary software. Accordingly, we are currently working on techniques to generate a watermark in the scanning algorithm that will uniquely identify an NIC.

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. Wright, J.: Detecting wireless LAN MAC address spoofing. http://home.jwu.edu/jwright/papers/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, www.wimetrics.com
12. iPass. http://www.ipass.com/services/servicesdeviceid.html
13. Cellular companies fight fraud. http://www.decodesystems.com/mt/97dec/
14. Hall, J.: Barbeau, M., Kranakis, E.: Detection of transient in radio frequency fingerprinting using signal phase. Internet and Information Technology (CIIT), St. Thomas, US Virgin Islands (2004)
15. Kohno, T., Briodo, A., Claffy, K.C.: Remote physical device fingerprinting. IEEE Trans. Dependable Secure Comput. **2**(2), 93–108 (2005)
16. Corbett, C., Beyah, R., Copeland, J.: A passive approach to wireless NIC identification. To appear in the Proceedings of IEEE International Conference on Communications (ICC) (2006)
17. Fyodor, Y.: Remote OS detection via TCP/IP stack fingerprinting. October 18, 1998. http://www.insecure.org/nmap/nmap-fingerprinting-article.txt
18. Arkin, O., Yarochkin, F.: Xprove v2.0: a fuzzy approach to remote active operating system fingerprinting. August 2, 2002. http://www.sys-security.com/archive/papers/Xprobe2.pdf
19. Agere's WiFi chipset reaches 150Mbit/s. www.electronicsweekly.com/Article5144.html
20. IEEE 802.11 specification, http://standards.ieee.org/getieee802/802.11.html
21. Mishra, A., Shin, M., Arbaugh, W.: An empirical analysis of the IEEE 802.11 MAC layer handoff process. ACM Comput Commun. Rev. **33**(2), 93–102 (2003)
22. Ramani, I., Savage, S.: SyncScan: practical fast handoff for 802.11 infrastructure networks. In: Proceedings of IEEE INFOCOM (2005)
23. Cheng, C.-M., Kung, H.T., Tan, K.-S.: Use of spectral analysis in defense against DoS attacks. In: Proceedings of the IEEE GLOBECOM, Taipei, Taiwan (2002)
24. Hussain, A., Heidemann, J., Papadopoulos, C.: Identification of repeated attacks using network traffic forensics. Technical Report ISI-TR-2003–577b, USC/Information Sciences Institute (2003)
25. Partridge, C. et al.: Using signal processing to analyze wireless data traffic. ACM Workshop on Wireless Security (WiSe), Atlanta, GA, USA, September 28, (2002)
26. McClellan, J., Schafer, R., Yoder, M.: Signal processing first. Prentice Hall, New York (2003)
27. Oppenheim, A.V., Schafer, R.W.: Discrete-time signal processing. pp. 730–742. Prentice-Hall, New York (1989)
28. Signal Processing Toolbox. http://www.mathworks.com/access/helpdesk/help/toolbox/signal/
29. The linux-wlan^{TM} Project. http://www.linux-wlan.org/
30. http://www.tcpdump.org/