802.11 User Fingerprinting

Jeffrey Pang[*] Ben Greenstein[†] Ramakrishna Gummadi[‡]
Srinivasan Seshan[*] David Wetherall[†§]

[*]Carnegie Mellon University [†]Intel Research Seattle
[‡]University of Southern California [§]University of Washington

jeffpang@cs.cmu.edu benjamin.m.greenstein@intel.com gummadi@usc.edu
srini@cmu.edu djw@cs.washington.edu

ABSTRACT

The ubiquity of 802.11 devices and networks enables anyone to track our every move with alarming ease. Each 802.11 device transmits a globally unique and persistent MAC address and thus is trivially identifiable. In response, recent research has proposed replacing such identifiers with pseudonyms (i.e., temporary, unlinkable names). In this paper, we demonstrate that pseudonyms are insufficient to prevent tracking of 802.11 devices because *implicit identifiers*, or identifying characteristics of 802.11 traffic, can identify many users with high accuracy. For example, even without unique names and addresses, we estimate that an adversary can identify 64% of users with 90% accuracy when they spend a day at a busy hot spot. We present an automated procedure based on four previously unrecognized implicit identifiers that can identify users in three real 802.11 traces even when pseudonyms and encryption are employed. We find that the majority of users can be identified using our techniques, but our ability to identify users is not uniform; some users are not easily identifiable. Nonetheless, we show that even a single implicit identifier is sufficient to distinguish many users. Therefore, we argue that design considerations beyond eliminating explicit identifiers (i.e., unique names and addresses), must be addressed in order to prevent user tracking in wireless networks.

Categories and Subject Descriptors:
C.2.1 Computer-Communication Networks: Network Architecture and Design

General Terms: Measurement, Security

Keywords: privacy, anonymity, wireless, 802.11

1. INTRODUCTION

The alarming ease with which third parties can track our every move has drawn the concern of the popular media [1, 2], the United States government [22, 40], and technical standards bodies [17]. The fear is that we are sacrificing our *location privacy* due to the ubiquity of wireless devices that disclose our locations, identities, or both. Though this fear has focused on large scale wireless systems, such as cellular phone networks, the capability to track user location in such systems has typically been limited to service providers that are legally bound to protect our privacy. In contrast, the low cost of 802.11 hardware and ease of access to network monitoring software—all that is required for someone to locate others nearby and eavesdrop on their traffic—enable *anyone* to track users. Furthermore, although the popular press raised awareness about tracking threats posed by emerging wireless technologies, such as RFID [13], no such campaign has been waged to educate users about 802.11 devices and networks, which pose the same threats *today*.

The best practices for securing 802.11 networks, embodied in the 802.11i standard [16], provide user authentication, service authentication, data confidentiality, and data integrity. However, they do not provide anonymity, a property essential to prevent location tracking. For example, it is trivial to track an 802.11 device today since each device advertises a globally unique and persistent MAC address with every frame that it transmits. To mask this identifier, researchers have proposed applying *pseudonyms* [14, 18] (i.e., temporary, unlinkable names) by having users periodically change the MAC addresses of their 802.11 devices.

In this paper, we demonstrate that pseudonyms are insufficient to provide anonymity in 802.11. Even without a unique address, characteristics of users' 802.11 traffic can identify them implicitly and track them with high accuracy. An example of such an *implicit identifier* is the IP address of a service that a user frequently accesses, such as his or her email server. In a population of several hundred users, this address might be unique to one individual; thus, the mere observation of this IP address would indicate the presence of that user. Of course, in a wireless network that employs link-layer encryption, IP addresses would not be visible to an eavesdropper. However, other implicit identifiers would remain and these identifiers can be used in combination to identify users accurately.

This paper quantifies how well a passive adversary can track users with four implicit identifiers visible to commodity hardware. We thereby place a *lower bound* on how accurately users can be identified implicitly, as more implicit identifiers and more capable adversaries exist in practice. We make the following contributions:

- We identify four previously unrecognized implicit identifiers: network destinations, network names advertised in 802.11 probes, differing configurations of 802.11 options, and sizes of broadcast packets that hint at their contents.

- We develop an automated procedure to identify users. This procedure allows us to quantify how much information implicit identifiers, both alone and in combination, reveal about several hundred users in three empirical 802.11 traces.

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.
Copyright 2007 ACM 978-1-59593-681-3/07/0009 ...$5.00.

- Our evaluation shows that users emit highly discriminating implicit identifiers, and, thus, even a small sample of network traffic can identify them more than half (56%) of the time in public networks, on average. Moreover, we will almost never mistake them as the source of other network traffic (1% of the time). Since adversaries will obtain multiple traffic samples from a user over time, this high accuracy in traffic classification enables them to track many users with even higher accuracy in common wireless networks. For example, an adversary can identify 64% of users with 90% accuracy when they spend a day at a busy hot spot that serves 25 concurrent users each hour.
- To our knowledge, we are the first to show with empirical evidence that design considerations beyond eliminating explicit identifiers, such as unique names and addresses, must be addressed to protect anonymity in wireless networks.

In Section 2 we illustrate the power of implicit identifiers with several real examples. Section 3 covers related work. Section 4 explains our experimental methodology. Section 5 describes our empirical 802.11 traces. Section 6 analyzes how well 802.11 users can be identified using each implicit identifier individually. Section 7 examines how accurately an adversary can track people using these implicit identifiers in public, home, and enterprise networks. We conclude in Section 8.

2. THE IMPLICIT IDENTIFIER PROBLEM

How significantly do implicit identifiers erode location privacy? Consider the seemingly innocuous trace of 802.11 traffic collected at the 2004 SIGCOMM conference, now anonymized and archived for public use [31]. Interestingly, hashing real MAC addresses to pseudonyms is also the best practice for anonymizing traces such as this. Unfortunately, implicit identifiers remain and they are sufficient to identify many SIGCOMM attendees. For example:

Implicit identifiers can identify us uniquely. One particular attendee's laptop transmitted requests for the network names "MIT," "StataCenter," and "roofnet," identifying him or her as someone probably from Cambridge, MA. This occurred because the default behavior of a Windows laptop is to actively search for the user's preferred networks by name, or Service Set Identifier (SSID). The SSID "therobertmorris" perhaps identifies this person uniquely [26]. A second attendee requested "University of Washington" and "djw." The last SSID is unique in the SIGCOMM trace and suggests that this person may be University of Washington Professor David J. Wetherall, one of our coauthors. More distressingly, Wigle [39], an online database of 802.11 networks observed around the world, shows that there is only one "djw" network in the entire Seattle area. Wigle happens to locate this network within 192 feet of David Wetherall's home.

Implicit identifiers remain even when counter measures are employed. Another SIGCOMM attendee transferred 512MB of data via BitTorrent (this user contacted hosts on the typical BitTorrent port, 6881). A request for the SSID "roofnet" [32] from the same MAC address suggests that this user is from Cambridge, MA. Suppose that this user had been more stealthy and changed his or her MAC address periodically. In this particular case, since the user had not requested the SSID during the time he or she had been downloading, the MAC address used in the SSID request would have been different from the one used in BitTorrent packets. Therefore, we would not be able to use the MAC address to explicitly link "roofnet" to this poor network etiquette. However, the user does access the same SSH and IMAP server nearly every hour and was the only user at SIGCOMM to do so. Thus, this server's address is an implicit identifier, and knowledge of it enables us to link the user's sessions together.

Now suppose that the network employed link-layer encryption scheme, such as WPA, that obscures network addresses. Even then, we could link this user's sessions together by employing the fact that, of the 341 users that sent 802.11 broadcast packets, this was the only one that sent broadcast packets of sizes 239, 245, and 257 bytes and did so repeatedly throughout the entire conference. Furthermore, the identical 802.11 capabilities advertised in each session's management frames improves our confidence of this linkage because these capabilities differentiate different 802.11 cards and drivers. Prior research has shown that peer-to-peer file sharing traffic can be detected through encryption [42]. Thus, even if pseudonyms and link-layer encryption were employed, we could still implicate someone in Cambridge.

Implicit identifiers are exposed by design flaws. These examples illustrate three shortcomings of the 802.11 protocol beyond exposing explicit identifiers, none of which is trivially fixed. These shortcomings afflict not only 802.11 but many wireless protocols, including Bluetooth and ZigBee.

Identifying information exposed at higher layers of the network stack is not adequately masked. For example, even with encryption, packet sizes can be identifying. Padding, decoy transmissions, and delays may hide information exposed by size and timing channels, but increase overhead. For example, Sun *et al.* [34] found that 8 to 16 KB of padding is required to hide the identity of web objects. The performance penalty due to this overhead would be especially acute in wireless networks due to shared nature of the medium.

Identifying information during service discovery is not masked. 802.11 service discovery can not be encrypted since no shared keys exist prior to association. This raises the more general problem of how two devices can discover each other in a private manner, which is expensive to solve [4]. This problem arises not only when searching for access points, but also when clients want to locate devices in ad hoc mode, such as when using a Microsoft Zune to share music or a Nintendo DS to play games with friends.

Identifying information exposed by variations in implementation and configuration is not masked. Each 802.11 implementation typically supports different 802.11 features (e.g., supported rates) and has different timing characteristics. This problem is difficult to solve due to the inherent ambiguity of human specifications and manufacturers' and network implementers' desire for flexibility to meet differing constraints.

Balancing the costs involved in rectifying these shortcomings with the incentives necessary for deployment is itself a challenge. Nonetheless, rectifying these flaws at the protocol level is important so that users need not limit their activities in order to protect their location privacy. By measuring the magnitude with which each flaw contributes to the implicit identifier problem, our study provides insight into the proper trade-offs to make when correcting these design flaws in future wireless protocols. In the short term, our study may give guidance to individuals that are willing to proactively hide their identity in existing wireless networks.

In the remainder of this paper, we examine how these shortcomings impact the location privacy of a large number of users in different 802.11 networks and demonstrate that the examples described in this section are not isolated anomalies.

3. RELATED WORK

The challenge of hiding a user's identity has been examined in three different contexts: location privacy, identity hiding designs, and the study of other implicit identifiers. In this section, we describe the previous work in each of these areas.

Location Privacy. Location privacy has recently received significant attention, most notably in the RFID [13] and pervasive computing [7] fields. The concern is that location-aware applications, which use GPS and other positioning technologies, might reveal this information in undesirable ways. However, location privacy is threatened even by devices that do not explicitly track location. Since 802.11 users usually associate with access points that are less than tens of meters away, knowing the access point that a user is associated with gives away a coarse estimate of his location, such as his home or workplace. Moreover, systems that can employ multiple monitoring locations can use wireless signal strength to obtain an even more accurate estimate of a user's location [6, 35]. An added complication is that wireless devices are rapidly becoming integral parts of our daily lives. A resulting trend, which is evident from examining databases of access point locations [39], is the increasing availability of service, which is increasing the number of location tracking opportunities. Unfortunately, identifying individual users is often trivial since the 802.11 devices that they use are uniquely named by their MAC addresses.

Identity Hiding. Pseudonyms are widely used in systems, such as the GSM cellular phone network [15] to hide user identities. Gruteser *et al.* [14] and Jiang *et al.* [18] proposed using pseudonyms within 802.11 networks, and Stajano *et al.* [41] proposed a similar mechanism for Bluetooth. Using pseudonyms is a necessary first step to make tracking in these networks more difficult. However, we show that it is insufficient to protect location privacy because *implicit identifiers* can be sufficient to track users in many real scenarios.

Implicit Identifiers. Fingerprinting devices using implicit identifiers is not a new concept. For example, Franklin *et al.* [11] showed that it is possible to fingerprint device drivers using the timing of 802.11 probes. In contrast, our work attempts to pin down actual user identities rather than selecting among a few dozen drivers.

Kohno *et al.* [21] showed that devices could be fingerprinted using the clock skew exposed by TCP timestamps. We introduce new implicit identifiers that are useful in identifying users and, in contrast to TCP timestamps, three of our identifiers are still visible in wireless networks using link-layer encryption. Moreover, Kohno *et al.* note that one limitation of their work is that an adversary can not passively obtain timestamps from devices running the most prevalent operating system, Windows XP. For example, in two of our empirical traces, only 32% and 15% of the users sent TCP timestamps. All our identifiers have much at least 55% coverage.

Padmanabhan and Yang [29] explored fingerprinting users with "clickprints," or the paths that users take through a website. Their techniques rely on data from many user sessions collected at actual web servers. Our techniques can be employed passively by anyone with a wireless card without even associating to a network. These three research efforts compliment ours, since the procedure we develop for identifying users enables an adversary to use these implicit identifiers in combination with ours, yielding even more accurate user fingerprints. None of these previous efforts offer a formal method to combine multiple pieces of evidence. Moreover, to our knowledge, we are the first to evaluate the how well users are identified by implicit identifiers observed in empirical wireless data.

Implicit identifiers also reveal identity in other contexts. Security tools like `nmap` [12] and `p0f` [28] leverage differences in network stack behaviors to determine a device's operating system. Keystroke dynamics have been shown to accurately identify users [24, 33]. The timing and sizes of Web transfers often uniquely identify websites, even when transmitted over encrypted channels [8, 34]. Finally, there has been a large body of research in identifying applications from implicit identifiers in encrypted traffic [19, 20, 25, 42, 43]. Like many of these techniques which succeed in classifying applications accurately, we use a Bayesian approach.

4. EXPERIMENTAL SETUP

This section describes the evaluation criteria we use to determine how well several implicit identifiers can be used to track users.

The Adversary. Strong adversaries, such as service providers and large monitoring networks, obviously pose a large threat to our location privacy. However, the significance of the threat posed by 802.11 is that *anyone* that wishes to track users can do so.

Therefore, we consider an adversary that runs readily available monitoring software, such as `tcpdump` [37], on one or more laptops or on less conspicuous commodity 802.11 devices [3]. We further restrict adversaries by assuming that their devices listen passively. That is, they never transmits 802.11 frames, not even to associate with a network. This means that the adversary *can not be detected* by other radios. The adversary deploys monitoring devices in one or more locations in order to observe 802.11 traffic from nearby users. By considering a weak adversary, we place a lower bound on the accuracy with which users can be tracked, as stronger adversaries would be strictly more successful.

The Environments. An adversary's tracking accuracy will depend on the 802.11 networks he or she is monitoring. Since implicit identifiers are not perfectly identifying, it will be more difficult to distinguish users in more populous networks. In addition, different networks employ different levels of security, making some implicit identifiers invisible to an adversary. We consider the three dominant forms of wireless deployments today: public networks, home networks, and enterprise networks.

Public networks, such as hot spots or metro-area networks [27], are typically unencrypted at the link-layer. Although many public networks employ access control—for example, to allow access to only a provider's customers—most do so via authentication above the link-layer (e.g., through a web page) and by using MAC address filtering thereafter. Very few use 802.11i-compliant protocols that also enable encryption. Hence, identifying features at the network, link, and physical layers would be visible to an eavesdropper in such an environment. Unfortunately, this is the most common type of network today due to the challenge of secure key distribution.

Home and small business networks are small, but detecting when specific users are present is increasingly challenging due to the high density of access points in urban areas [5]. In addition, these networks are more likely to employ link-layer encryption, such as WEP or WPA, because the set of authorized users is typically known and is small. In cases where link-layer encryption is employed, an eavesdropper will not be able to view the payloads of data packets. However, features that are derived from frame sizes or timing, which are not masked by encryption, or from 802.11 management frames, which are always sent in the clear, remain visible.

Finally, security conscious enterprise networks are likely to employ link-layer encryption. Moreover, if the only authorized devices on the network are provided by the company, there will be less diversity in the behavior of wireless cards. For example, Intel corporation issues similar corporate laptops to its employees. We consider a enterprise network where only one type of wireless card and configuration is in use, so users can not be identified by differences in device implementation. However, features derived from the networks that users visit or the applications and services they run remain visible.

The Monitoring Scenario. We assume that users use different pseudonyms during each wireless session in each of these environments, as Gruteser *et al.* [14] propose. As a result, explicit identifiers can not link their sessions together. Sessions can vary in length, so we assume that every hour, each user will have a different pseudonym. We define a *traffic sample* to be one user's network traffic observed during one hour.

Although it is possible for users to change their MAC addresses more frequently, this is unlikely to be very useful in practice because other features, such as received signal strength, can link pseudonyms together at these timescales [6, 35]. Moreover, changing a device's MAC address forces a device to re-associate with the access point and, thus, disrupts active connections. In addition, it may require users to revisit a web page to re-authenticate themselves, since MAC addresses are tied to user accounts in many public networks. Users are unlikely to tolerate these annoyances multiple times per session.

Of course, the ability to link traffic samples together does not help an adversary detect a user's presence unless the adversary is also able to link at least one sample to that user's identity. In Section 2, we showed that identity can sometimes be revealed by correlating implicit identifiers with out-of-band information, such as that provided by the Wigle [39] location database. However, if the adversary knows the user he wishes to track, he can likely obtain a few traffic samples known to come from that user's device. For example, an adversary could obtain such samples by physically tracking a person for a short time. We assume the adversary is able to obtain this set of *training samples* either before, during, or after the monitoring period. Our results show that on average, only 1 to 3 training samples are sufficient to track users with each implicit identifier (see Section 6.2.3). The monitor itself collects samples that the adversary wants to test, which we call *validation samples*.

Evaluation Criteria. There are a number of questions an adversary may wish to answer with these validation samples. Who was present? When was user U present? Which samples came from user U? Essential to answering all these questions is the ability to classify samples by the user who generated them. In other words, given a validation sample, the adversary needs to answer the following question for one or more users U:

Question 1 *Did this traffic sample come from user U?*

Section 6 evaluates how well an adversary can answer this question with each of our implicit identifiers.

To demonstrate how well implicit identifiers can be used for tracking, we also evaluate the accuracy in answering the following:

Question 2 *Was user U here today?*

This question is distinct from Question 1 because an adversary can observe many traffic samples at any given time, any one of which may be from the target user U. In addition, a single affirmative answer to Question 1 does not necessitate a affirmative answer to Question 2 because an adversary may want to be more certain by obtaining multiple positive samples. Section 7 details the interaction between these questions and evaluates how many users can be tracked with high accuracy in each of the 802.11 networks described above.

5. WIRELESS TRACES

We evaluate the implicit identifiers of users in three 802.11 traces. We consider `sigcomm`, a 4 day trace taken from one monitoring point at the 2004 SIGCOMM conference [31], `ucsd`, a trace of all 802.11 traffic in U.C. San Diego's computer science building on November 17, 2006 [10], and `apt`, a 19 day trace monitoring all networks in an apartment building, which we collected. All traces were collected with `tcpdump`-like tools and only contain information that can be collected using standard wireless cards in monitor mode. The `ucsd` trace is the union of observations from multiple monitoring points. IP and MAC addresses are anonymized but are consistent throughout each trace (i.e., there is a unique one-to-one mapping between addresses and anonymized labels). Link-layer encryption (i.e., WEP or WPA) was not employed in either the `sigcomm` or `ucsd` network and neither trace recorded application packet payloads. In our analysis, we show that implicit identifiers remain even when we emulate link layer encryption and that we do not need packet payloads to identify users accurately. The `apt` trace only recorded broadcast management packets due to privacy concerns; hence, we only use it to study the one implicit identifier that is extracted from these packets.

We distinguish unique users by their MAC address since it is not currently common practice to change it. To simulate the effect of using pseudonyms, we assume that every user has a different MAC address each hour. Hence, we have one sample per user for each hour that they are active. To simulate the training samples collected by an adversary, we split each trace into two temporally contiguous parts. Samples from the first part are used as training samples and the remainder are validation samples. We choose a training period in each trace long enough to profile a large number of users. For the `sigcomm` trace, the training period covers the time until the end of the first full day of the conference. For the `ucsd` trace, the training period covers the time until just before noon. We skip one hour between the training and validation periods so user activities at the end of the training period are less likely to carry over to the validation period. For the `apt` trace, the training period covers the first 5 days. We consider a user to be present during an hour if and only if she sends at least one data or 802.11 probe packets during that time; i.e., if the user is actively using or searching for a wireless network.[1]

Table 1 shows the relevant statistics about each trace. Note that since can we only compute accuracy for users that were present in both the training and validation data, those are the only users that we profile. Therefore, results in this paper refer to 'Profiled Users' as the total user count and not 'Total Users.'

6. IMPLICIT IDENTIFIERS

In this section, we describe four novel implicit identifiers and evaluate how much information each one reveals. Our results show that (1) many implicit identifiers are effective at distinguishing individual users and others are effective at distinguishing groups of users; (2) a non-trivial fraction of users are trackable using any one highly discriminating identifier; (3) on average, only 1 to 3 training samples are required to leverage each implicit identifier to its full effect; and (4) at least one implicit identifier that we examine accurately identifies users over multiple weeks.

[1] We ignore samples that only contain other 802.11 management frames, such as power management polls. Including samples with these frames would not appreciably change the characteristics of the `sigcomm` workload, but would double the number of total "users" in the `ucsd` workload. This is because many devices observed in the `ucsd` trace were never actively using the network; we ignore these idle devices.

	sigcomm		ucsd		apt	
	training	validation	training	validation	training	validation
Duration (hours)	37	54	10	11	119	345
Total Samples	1974	3391	587	1240	638	1473
Frames Per Sample (median)	289	284	1227	1128	57	92
Total Users	377	412	225	371	97	196
Profiled Users	337	337	153	153	39	39
Samples Per Profiled User (mean)	5.5	9.1	3.1	4.7	14.7	32.2
Users Per Hour (mean)	53	64	59	113	5	4

Table 1—Summary of relevant workload statistics and parameters. The duration reports only hours with at least one active user.

6.1 Identifying Traffic Characteristics

Network Destinations. We first consider netdests, the set of IP <address, port> pairs in a traffic sample, excluding pairs that are known to be common to all users, such as the address of the local network's DHCP server. There are several reasons to believe that this set is relatively unique to each user. It is well known that the popularity of web sites has a Zipf distribution [9], so many sites are visited by a small number of users. In fact, in the sigcomm and ucsd training data, each <address, port> pair is visited by 1.15 and 1.20 users on average, respectively. The *set* of sites that a user visits is even more likely to be unique. In addition, users are likely to visit some of the same sites repeatedly over time. For example, a user generally has only one email server and a set of bookmarked sites they check often [36].

An adversary could obtain network addresses in any wireless network that does not enable link layer encryption. Even if users sent all their traffic through VPNs, the case for several users in the sigcomm trace, the IP addresses of the VPN servers would be revealing. No application or network level confidentiality mechanisms, such as SSL or IPSec, would mask this identifier either.

SSID Probes. Next we consider ssids, the set of SSIDs in 802.11 probes observed in a traffic sample. Windows XP and OS X add the SSID of a network to a preferred networks list when the client first associates with the network. To simplify future associations, subsequent attempts to discover *any* network will try to locate this network by transmitting the SSID in a probe request. As we observed in Section 2, SSID names can be distinguishing.[2] In addition, probes are never encrypted because active probing must be able to occur before association and key agreement.

There are two practical issues that limit the use of ssids as an implicit identifier. First, the preferred networks list changes each time a user adds a network, and thus a profile may degrade over time. Second, clients transmit the SSIDs on their preferred networks lists only when attempting to discover service. Therefore, clients may not probe for distinguishing SSIDs very often. While this is true, our results show that when distinguishing SSIDs are probed for, they can often uniquely identify a user. Since all users in the monitoring area are likely to use the SSIDs of the networks being monitored, these SSIDs are not distinguishing and we do not include them in the ssids set.

Broadcast Packet Sizes. We now consider bcast, the set of 802.11 broadcast packet sizes in each traffic sample. Many applications broadcast packets to advertise their existence to other machines on the local network. Due to the nature of this function, these packets often contain naming information. For example, in our traces, we observed many Windows machines broadcasting NetBIOS naming advertisements and applications such as FileMaker and Microsoft Office advertising themselves.

Application	Port	Number of Sizes
wireless driver or OS	NA	14
DHCP	67	14
sunrpc	111	1
NetBIOS	138	7
groove-dpp	1211	1
Microsoft Office v.X	2222	1
FileMaker Pro	5003	7
X Windows	6000	1

Table 2—A list of the most unique broadcast packets observed in the sigcomm trace. The third column shows the number of packet sizes that were emitted by at most 2 users.

Since these packets vary in length, their sizes can reveal information about their content even if the content itself is encrypted. Packet sizes alone appear to distinguish users almost as well as <application, size> tuples. For example, in the sigcomm trace, there are only 16% more unique tuples than unique sizes. Table 2 lists the most unique broadcast packet sizes we observed and the application port that generated them. Broadcast packets are sent to a known broadcast MAC address; thus, an adversary can distinguish them from other traffic even if link encryption is employed and the adversary is not granted network privileges. This set would remain identifying even when user behavior changes because most broadcast packets are emitted automatically.

Two types of broadcast packets, standard DHCP requests and power management beacons, are common to all users, since a device must send a DHCP request in order to obtain an IP address and sends power management beacons when in low power mode. Thus, we do not include these packets' sizes in the bcast set. These packets have distinct sizes (336 and 36 payload bytes, respectively) so they can be filtered even when link-layer encryption is enabled.

MAC Protocol Fields. Finally, we consider fields, the specific combination of 802.11 protocol fields visible in the MAC header that distinguish a user's wireless card, driver, and configuration. The fields included are the 'more fragments,' 'retry,' 'power management,' and 'order,' bits in the header, the authentication algorithms offered, and the supported transmission rates. Some card configurations can be more or less likely to emit different values in each of these fields, so they can distinguish users with different wireless cards. Although this identifier is unlikely to distinguish users uniquely, it can be combined with others to add more evidence. Moreover, many of these fields are available in any 802.11 packet, so they can almost always assist in identification. Furthermore, the likelihood of any particular field combination is unlikely to change for a user unless she obtains a new wireless device or

[2] A recent patch [23] to Windows XP allows a user to disable active probing, but it remains enabled by default because disabling it would break association in networks where the access point does not announce itself. In addition, revealing probes or beacons are still required for devices to discover each other in ad hoc mode.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.