

Disambiguation of Residential Wired and Wireless Access in a Forensic Setting

Sookhyun Yang Jim Kurose Brian Neil Levine
Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA, 01003
{shyang, kurose, brian}@cs.umass.edu

Abstract—Thousands of cases each year of child exploitation on P2P file sharing networks lead from an IP address to a home. A first step upon execution of a search warrant is to determine if the home’s open Wi-Fi or the closed wired Ethernet was used for trafficking; in the latter case, a resident user is more likely to be the responsible party.

We propose methods that use remotely measured traffic to disambiguate wired and wireless residential medium access. Our practical techniques work across the Internet by estimating the per-flow distribution of inter-arrival times for different home access network types. We observe that the change of inter-arrival time distribution is subject to several residential factors, including differences between OS network stacks, and cable network mechanisms. We propose a model to explain the observed patterns of inter-arrival times, and we study the ability of supervised learning classifiers to differentiate between wired and wireless access based on these remote traffic measurements.

I. INTRODUCTION

Images of child sexual exploitation are common on BitTorrent, Gnutella, and other file-sharing networks [6], [11]. The end result of network-based investigations of these crimes is evidence that supports a court-issued warrant to enter and search the home associated with the observed IP address. A common alibi is that a third party used the home’s open Wi-Fi. A useful first step during execution of the warrant would be to determine if contraband was distributed on the P2P network using the house’s Ethernet network, therefore making the resident user more likely to be the responsible party, or if Wi-Fi was used and the alibi is justified. Indeed, drive-by abuse of open Wi-Fi by criminals has been a documented practice for years [5], [9], but methods to distinguish such access are unavailable.

This paper investigates methods that use remotely measured traffic to disambiguate wired and wireless residential medium access. Importantly, we place our work in a practical forensic setting by *constraining our approaches to only use remotely gathered “plain view” data that can be gathered legally from p2p networks before a warrant or wiretap is required (in the US)*. This constraint distinguishes our work from previous wired/wireless disambiguation research, which has assumed that measurements are taken from the target’s gateway router, which is not only a much less challenging problem but impractical from a forensic setting since it violates the Wiretap Act. Our goal is to provide information to investigators as they execute a search warrant inside a home. In addition to checking alibis and supplying information for a suspect’s interview, our techniques are also useful for *forensic triage*. Backlogs of six months are typical for criminal forensics labs, and the easiest way to

reduce the queue is to exclude computers from consideration (for example, those that have no wired interface) [8].

Our techniques work across the Internet by estimating the per-flow distribution of *inter-arrival times* of packets transmitted over different types of home access networks, as measured by an investigator at a remote Internet P2P client. Using a set of traces that we collected, we evaluate the ability of a number of classifiers to *remotely* distinguish wired from wireless access within the same house. We find that our approach for classifying wired from wireless traffic can work well, but is subject to several residential factors such as OS network stacks, and cable modem mechanisms. Our analysis reveals the following:

- We use a simple decision tree classifier that uses remotely measured traces and identify **25th percentiles** or **entropy** of inter-arrival times distribution of the traces as classification features, achieving a true positive rate (TPR) of 0.9 to 1.0 and false positive rate (FPR) of 0.0 to 0.1 in our studies. For Linux, we can precisely classify wired from wireless using 25th percentiles or entropy in accordance with a cable network’s state. But for Windows, we can depend on only entropy as a good classification feature.
- We evaluate the cases of single and multiple P2P flows from the source, but we find that this distinction does not affect our result. Only the individual throughput of each flow has an impact on the classifier.
- Our classifier must be trained separately for different throughputs from the target; fortunately, this throughput is easily observable at the receiver. Such training can be performed when the search warrant is executed from within the house; there is no reason to train a general classifier for all houses ahead of the warrant’s execution.

Overall, our findings suggest that it is difficult at best to find a foolproof classifier for remote identification in all scenarios. Our goal is to determine the scenarios in which network access type can be accurately determined, and to understand when these techniques cannot be reliably used in other scenarios. Additional results beyond those reported in this paper can be found in [15].

II. INVESTIGATIVE METHOD AND JUSTIFICATION

The general criminal procedure for child pornography (CP) cases is as follows.

1) Investigators search for content on P2P networks. 2) CP files offered in plain view by a peer, identified by IP address,

are downloaded by investigators. 3) The download provides sufficient *probable cause* as part of an application for a magistrate-issued search warrant of the home associated with the IP address’s billing records. 4) The warrant is issued, and once inside the home, a triage-style search begins for evidence associated with CP, which might not be the previously downloaded content. Users of the home’s computers are interviewed. 5) Seized devices are sent to an off-site lab for detailed forensic examination. 6) Evidence found during search is then used to support a criminal trial for receipt, possession, or distribution of CP.

The step of searching a home is time consuming. Homes have an increasing number of devices that can contain evidence, including small devices with web-browsers, desktops, and laptops. Investigators have three main *triage* aims: *a)* reducing the numbers of devices that must be examined on-scene since warrants are time-limited; *b)* reducing the number of devices that must be sent to an off-site central forensics lab for in-depth examination since work queues are months-long; and *c)* quickly locating a subset of evidence, if it exists, so as to obtain an admission of guilt by a suspect via an interview. All these practical goals are met more efficiently by knowing whether a computer used over the Internet is likely wired or wireless.

Our goal is to examine whether it is possible to remotely infer the target’s access type. Our technique would be used as follows. During Step 1) above, investigators would keep a packet-level trace of the file download, which is already common practice. Using the packet-level trace, investigators identify a criminal’s computer setting (such as OS network stacks and P2P applications), and characterize the download throughput and *concatenation* rate (as we will see in Section V). This information is not a part of the warrant application. During Step 3), the classifier is trained, which can be completed in minutes with a pre-configured program and a laptop with both wired and wireless interfaces. The pre-configured program regenerates the flow, having equivalent throughput and concatenation rate observed in Step 1) via wired and wireless interfaces. The results of classification tests are used on scene to inform triage and user interviews. We note that it would only reduce accuracy to pre-train a classifier from general Internet scenarios.

Importantly, our collection takes place at the investigator’s end host. This measurement is possible without warrant or wiretap since the investigator is a party to the communication. In contrast, previous work proposes to collect packets at a network gateway, which is illegal in our forensics context. It is also impractical as investigators cannot know which gateway until they have a suspect; going back to the gateway after the suspect has uploaded the CP to the investigator is too late.

A number of legal issues restrict the initial process of gathering the data we use to infer a target’s medium access type [2], [4]. First, US law prohibits government search and seizure of evidence without a warrant if and only if the source of the data has a *reasonable expectation of privacy* (REP) [4]. US courts have found consistently that users of P2P file sharing networks have no REP when investigators are peers in the network; see *U.S. v. Breese*, 2008 WL 1376269 and *U.S. v.*

Gabel, 2010 WL 3927697. Collecting information at a user’s gateway without a warrant is certainly illegal.

Second, prior to obtaining a warrant, law enforcement cannot use technology that is not in “general public use” to obtain information that would otherwise be unavailable. This restriction is a result of *Kyllo v. U.S.*, 533 U.S. 27 (2001). For example, recently the court ruled that software designed for law enforcement to monitor activity on P2P networks does not violate 4th Amendment protections since if it follows the protocol as any peer on the network does. Similarly, in *Massachusetts v. Karch* (2011), the court ruled that law enforcement programs that do not search the remote computer, but “merely gather and evaluate publicly available information with greater efficiency and with an eye toward obtaining evidence of criminal activity” do not violate *Kyllo*, even if the software itself is unavailable for general public use.

Related work, in Section VI, that has been motivated by network monitoring and measurement is also governed by several US federal laws. Sicker et al. [10] provide an excellent overview and discussion of these laws and their consequences for the network traffic measurement research community. Criminal investigations are not included in that analysis since they lack the *provider protection* motive, which is measurement with the aim of protecting the network infrastructure, e.g., detecting or characterizing network attacks. In monitoring settings, clients typically consent to monitoring by the provider as part of an acceptable use policy.

Information gathered in a criminal investigation ideally meets the standards of criminal trials (beyond a reasonable doubt). However, information that meets the *probable cause* (PC) standard used to issue search warrants is still useful. There is no quantification of PC by courts; often it is defined qualitatively as a “fair probability”; see *U.S. v. Sokolow*, 490 U.S. 1 (1989). We evaluate our work with these standards in mind by quantifying true and false positive rates. Finally, we note that we expect that the techniques we introduce in this paper are most useful as simple, practical information to inform the process of search and triage, as noted above, rather than as evidence.

III. PROBLEM STATEMENT

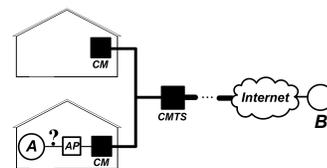


Fig. 1. An illustration of our expected network topology.

Our problem setting is illustrated in Fig. 1. As described in Section II, we begin by assuming that investigators have already identified a peer, denoted as *A* in the figure, who is a *target* that uploads illegal content to the investigator. *Our challenge is to determine whether A is connected to the home AP via a wireless 802.11 network or via a wired Ethernet.* Investigators, denoted as *B* in the figure, can make this determination using only traces measured from a remote location.

We assume the AP used by A is connected to the Internet via a cable modem (CM). The coordination system of a regional head-end, known as a Cable Modem Termination System (CMTS), regulates the use of upstream and downstream bandwidth based on A 's level of contracted service with the cable network service provider. The CM communicates with the CMTS using the Data Over Cable Service Interface Specification (DOCSIS) [1] protocol stack. In the downstream direction, the CMTS broadcasts data and control frames to a set of CMs. The upstream channel consists of a stream of time-slots shared among CMs. Using the DOCSIS, the CMTS replants to CM time-slots requests and grants time-slots to the CM using *MAP messages* every 2ms. Once the CM has acquired time-slots from the CMTS, it usually transmits a TCP segment per DOCSIS frame. In the case of congestion, the CM can buffer multiple TCP segments and concatenate the segments in a DOCSIS frame after waiting for a longer time-slot-granting delay. One manifestation of buffering at the CM has been recently noted in *bufferbloat* [3].

We consider measurements taken at B , where an investigator can legally make such measurements. B records the inter-arrival times of TCP data packets sent from A during file-sharing uploads to B . To provide the most general solution, we assume the investigator outside of the cable network performs measurement from a typical Internet end-point, and not a gateway router or other specialized device, and B has rich, high-speed connectivity to the Internet. A 's traffic will be transmitted through the cable network and then through a number of additional networks before arriving at B .

A. Factors Affecting TCP segment spacing and burst size

Since we will use the inter-arrival times between segments to distinguish between wired and wireless access in the sender's home, let us next consider how the TCP and DOCSIS protocols shape the time between transmission of A 's TCP segments. TCP's sliding window algorithm typically results in *bursts* of packets that are sent with only short inter-departure times between back-to-back segments. These bursts are then separated by a relatively longer interval of time, while the sender waits for the receiver's ACK.

When the CM transmits segments, the inter-departure time between two segments can be different from those segments' inter-arrival time to the CM. These changes can be small or significant, and can depend on the level of congestion in the cable network. Since the segments' inter-arrival times to the CM follow their departure from the (wired or wireless) access network to the CM, and since our goal is to distinguish between wired and wireless access times based on these inter-arrival times, we will focus on segments whose inter-departure time from the CM closely matches their inter-arrival time to the CM. In Section VI.A we present key insights that allow us to identify segments whose inter-departure time is relatively unchanged from their inter-arrival time.

Several other factors found in a typical setting also affect TCP burst sizes and segment inter-arrival times at the CM.

Multiple flows from A . A P2P peer often exchanges data with multiple peers simultaneously. Since upstream bandwidth is shared among these multiple flows, each individual flow will experience a lower throughput than in a single flow scenario. This decreased throughput is evidenced in a decreased burst size and increased inter-burst spacing. We find, however, that for accurate classification, we only need determine (by measurement) the throughput of a target flow; the number of competing flows need not be known.

TCP algorithms with its send buffer size. The TCP algorithms with its send buffer size play an important role in determining the burst size. Linux has a large maximum send buffer size and disables Nagle's algorithm by default. Consequently, Linux's burst sizes adaptively change over congestion window; conversely, Windows's burst size is often equal to its *very small* default send buffer size of 8 KB [7] due to Winsock buffering, which also makes segment inter-arrival times not affected by Nagle's algorithm.

In [15], we also consider P2P application rate limit and wireless channel contention as other factors to segment inter-arrival times.

IV. EXPERIMENTAL ENVIRONMENT AND METHODOLOGY

This section describes the experimental setting in which we obtained measurement traces. Our experiments do not include results from law enforcement trials. It would violate IRB protocols to experiment on Internet users without consent.

A. Experimental setting

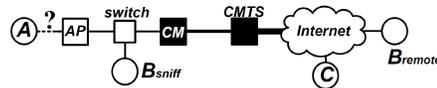


Fig. 2. Our measurement network topology for experiments.

Fig. 2 illustrates the experimental setting for our packet measurements. We have two monitoring points: B_{sniff} , and B_{remote} . Node A generates TCP traffic to B_{remote} using *iPerf* to transmit TCP data at the maximum rate possible; hence the TCP transmit queue is never starved for data. At B_{sniff} we place a sniffer that captures frames before they are transmitted via the CM. We stress that B_{sniff} is used here *only for experimental research purposes here; as discussed above, our practical forensic setting would preclude making measurements at this point in practice*. Our classification results are performed using *only* traces gathered at B_{remote} .

B_{remote} is located at the Univ. of Massachusetts Amherst. A is located in ten houses in a town near UMass Amherst, using Comcast's residential cable network. The number of hops between A and B_{remote} varies, as determined via *traceroute*. We used node C as the sink for TCP flows originating at A that compete with traffic to B . C was located at Purdue University. AP is the link type we seek to classify. A 's connection to AP was either IEEE 802.11g with 54 Mbps or 1 Gbps Ethernet in our study. For 802.11g measurement, we located A less than 1m away from the AP to obtain the wireless traces least distinguishable from wired traces. During wireless traffic generation, we also captured background wireless traffic via

another laptop’s *monitor mode* and the measured background traffic was commonly less than 1 Mbps.

The Comcast cable network supports DOCSIS v2.0 with 4 ticks per time-slot as an upstream modulation and 8,160 bytes as a maximum burst frame size. Thus, we can instantaneously see an upstream throughput up to 10 Mbps, although the upstream capacity of a contract is 3 Mbps.

We varied the experimental environment as follows.

Single flow vs. multiple flows. We separately evaluated cases of single and multiple competing TCP flows. In the single flow case, A generated a single TCP flow sent to B_{remote} . In the multiple flow case, A generated one TCP flow sent to B_{remote} and four competing TCP flows in parallel that were sent to C . These five flows generated from A were delivered to a single CM via either wired or wireless access.

Linux vs. Windows. Our traffic source was either Ubuntu with Linux Kernel 2.6.22 or Windows Vista at node A . Linux Kernel 2.6.22 uses CUBIC and Windows Vista uses CTCP. Since Kernel 2.4, Linux takes 4 KB as minimum, 16 KB as default, and 4 MB as maximum, and TCP dynamically adjusts the size of the send buffer based on these three values. Windows Vista has a TCP send buffer size of 8 KB by default; the same 8 KB default is also found in Windows XP and Windows 7.

Each above experiment setting was performed ten times for 10 sec, 1 min, or 10 mins. We captured a target flow via *tcpdump* at B_{sniff} and B_{remote} and then derived inter-arrival time datasets. We calculated the inter-arrival time as the time interval between the first bit of a first packet and the first bit of a second packet of two back-to-back TCP segments and considered only segments that experienced neither retransmission nor loss.

V. EVALUATION OF CLASSIFIER PERFORMANCE

In this section, we describe the experimental procedure for evaluating a *decision tree* (DT) classifier using the traces described in Section IV. We present our empirical evaluation of classification and verify our conjectures above regarding the circumstances in which different classifiers would work well.

A. Experimental procedure

For each experimental trace, we trained and cross-validated the classifier using datasets of wired and wireless traffic and investigated various features such as the 25th, 50th, 75th percentiles of the inter-arrival time distribution at the receiver, and the entropy of this distribution. We quantify classification accuracy using the *true positive rate* (TPR) and *false positive rate* (FPR). TPR denotes the fraction of cases where the access network type is classified as wired given that it is wired. FPR denotes the fraction of cases when the access network type is classified as wired given that it is actually wireless. If the TPR were to be low, the classifier would wrongly argue for accepting the false alibi of a wired user. If the FPR were to be high, the classifier would wrongly argue for not accepting a valid alibi (i.e., that the CP distributor actually did use a wireless network). For our purposes here, we consider it as an *acceptable result* when the TPR is located between 0.9 and 1 and the FPR is located between 0 and 0.1.

Based on our packet-spacing-model for residential traffic [15], we characterize a target TCP flow using burst size, throughput, and concatenation rate.

Burst size observed at B_{sniff} . α denotes the burstiness of a segment arrival process to the CM after leaving the host computer but before reaching the CM. Using the dataset measured at B_{sniff} , we calculate α as

$$\alpha = \left(\frac{\text{no. of inter-arrival times below 1ms at } B_{sniff}}{\text{total number of inter-arrival times at } B_{sniff}} \right).$$

In our setting, 802.11g uses neither the RTS/CTS option nor CTS protection but supports frame-burst. Since 802.11g spends $322\mu\text{s}$ on transmitting a full-sized TCP segment without random-backoff and frame-burst, most short inter-departure times in a burst are less than 1ms at B_{sniff} . *We stress α would not be used during a forensic investigation, nor do we employ it in our classification procedure. However, we will find it useful to use α to explain our classifier results, as α characterizes the burstiness of the (unobservable) source.*

Throughput observed at B_{remote} . We calculate an average A -to- B_{remote} throughput observed at B_{remote} (denoted by T). We will see that a flow with a lower throughput is more likely to have more good inter-arrival times than a flow with a higher throughput, and thus is more likely to result in more accurate classification. T is an important flow attribute to be considered in assessing the classification accuracy in Step 1). During Step 3), T would be used to generate a flow observed in step 1).

Concatenation rate observed at B_{remote} . A flow’s concatenation rate (denoted by β) is the fraction of segment inter-arrival times at B_{remote} that indicate that these two segments were concatenated in a single DOCSIS frame by the CM. We calculate β using the dataset measured at B_{remote} as

$$\beta = \left(\frac{\text{no. of inter-arrival times below 1ms at } B_{remote}}{\text{total number of inter-arrival times at } B_{remote}} \right).$$

A receiver can easily identify concatenated TCP segments as those segments having an inter-arrival time of less than 1ms since the CM must wait for at least 2ms to be granted a time slot from the CMTS. As we will see, the value of β will be an important factor in deciding whether to use percentiles or entropy values of the inter-arrival time distribution observed at B_{remote} as classification features.

The tables in the following subsections show an average of ten classification results for each experimental setting. The table only shows the traces from a single house, since our observations in the single house are consistent with what we observed in experiments run at other locations. Each dataset contained at least one thousand inter-arrival times. The inter-arrival times were produced by two successive full-sized (1,460 bytes) TCP segments. But approximately 30% of the inter-arrival times in the Windows with an 8 KB send buffer traces were observed to be transmissions of a burst of five full-sized segments followed by a 892-byte TCP segment. The tables show averages of α , β and T values of ten datasets for wired and wireless access networks. The rows in the tables show the TPR and FPR when we used the 25th-percentiles and the entropy of inter-arrival time distributions as features. Traces 1-3 and 5-7 show the results for a single full-rate flow with no concatenation (very low values of β) or higher concatenation.

Traces 4, and 8 show the results for multiple-flows.

B. Evaluation results

Linux	Trace 1		Trace 2		Trace 3		Trace 4	
	wired	wireless	wired	wireless	wired	wireless	wired	wireless
α	0.50	0.48	0.74	0.72	0.73	0.73	0.29	0.28
β	0.00	0.00	0.80	0.80	0.79	0.79	0.00	0.00
T (Mbps)	2.25 ± 0.05	2.25 ± 0.05	3.45 ± 0.05	3.45 ± 0.05	3.44 ± 0.01	3.44 ± 0.01	0.53 ± 0.00	0.54 ± 0.01
Features	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
25th-percentile	1.0*	0.1*	0.7	0.1	0.7	0.1	1.0*	0.0*
entropy	0.6	0.1	0.9*	0.0*	0.5	0.1	0.0	0.0

Windows	Trace 5		Trace 6		Trace 7		Trace 8	
	wired	wireless	wired	wireless	wired	wireless	wired	wireless
α	0.83	0.81	0.83	0.81	0.83	0.81	0.83	0.81
β	0.17	0.17	0.78	0.78	0.78	0.78	0.65	0.65
T (Mbps)	2.46 ± 0.05	2.46 ± 0.05	3.37 ± 0.05	3.37 ± 0.05	3.45 ± 0.05	3.45 ± 0.05	0.89 ± 0.01	0.86 ± 0.03
Features	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
25th-percentile	0.5	1.0	0.1	0.0	0.5	0.3	0.0	0.0
entropy	0.9	0.3	0.9*	0.1*	0.2	0.4	1.0*	0.1*

Let us begin our discussion of DT classification results by considering what we found to be the most difficult classification scenario as in [15]: classifying a single full-rate flow. Intuitively, we might expect this to be the most challenging case, since in this high throughput scenario, there are a large number of long bursts and minimal inter-burst-delay. In our discussions, we distinguish between Linux and Windows used at A , since sometimes our classification results will differ based on the OS type. Also, for the same send buffer size, we find that Windows and Linux can generate quite different values of α and β and that Windows consistently generates traffic with a non-negligible amount of concatenation.

Linux. Trace 1, a case with no concatenation, shows that the 25th-percentile classifier worked well but that the entropy classifier does not work well. Traces 2 and 3, cases with concatenation, show that the 25th-percentile classifier has a lower TPR (.7) and that neither the 25th-percentile classifier nor the entropy classifier work well in both traces 2 and 3 (although entropy works well as a classifier for trace 2). Compared with Traces 1-3, Trace 4 shows that classification of the multiple-flow scenario results in more accurate classification than classification of single full-rate flows.

Windows. Windows (with an 8 KB send buffer) consistently generated large bursts ($\alpha \approx 0.8$) as a result of Winsock buffering, resulting in the CM performing a mild degree of concatenation ($0.17 \leq \beta$), regardless of a cable network's congestion state. Consistent with our discussion in the previous section, we thus see that 25th-percentile classification does not work well for Windows. Additionally, we find that as with Linux, the entropy classifier does not work consistently well for single full-rate flows with high concatenation rates, but classification of the multiple-flow scenario is more accurate than classification of single full-rate flows.

In summary, we found that accurate classification of single full-rate flows is difficult with either classifier but multiple-flow scenarios show better results. Accurate and reliable classification of single full-rate flows remains an open challenge.

VI. RELATED WORK

Several past studies have addressed the problem of classifying a sender's access network type using traffic measurements.

Wei et al. [14] classified sender network access types into 802.11b, Ethernet, land ow-bandwidth wired access using cooperatively transmitted back-to-back UDP packet pairs between sender and receiver. Like our work, Wei et al. took measurements of packet inter-arrival times at the receiver. However, unlike our work, they assume UDP packet pairs are sent by a *cooperative sender*; instead we perform classification without the target's knowledge, using only (P2P application) TCP traffic, with the sender potentially engaging in multiple TCP sessions with multiple receivers. In subsequent work, Wei et al. [12], [13] monitored ACK packets exiting a university gateway and built a classifier for distinguishing between Ethernet and 802.11b/g traffic. Gateway measurement is not possible in our forensic setting, as this would violate the Wiretap Act. In contrast, we are focused on measurements taken from outside the source's network domain.

VII. CONCLUSIONS

We proposed legal methods that use remotely measured traffic to disambiguate wired and wireless residential medium access of a criminal in a practical forensic setting, leveraging the difference in inter-arrival times in the wired and wireless access networks. We justified our method's legality based on US law and extensively considered the effect of unknown or hidden factors in a forensic setting on classification performance. We identified 25th-percentile or entropy of inter-arrival times as the best performing features and figured out when these features worked reliably or poorly in diverse scenarios.

Acknowledgments: This research is supported by NSF awards CNS-0905349 and CNS-1040781.

REFERENCES

- [1] CableLabs. Data-Over-Cable Service Interface Specifications (DOCSIS).
- [2] E. Casey. *Digital evidence and computer crime: forensic science, computers and the Internet*. Academic Pr, 2004.
- [3] J. Gettys and K. Nichols. Bufferbloat: Dark buffers in the internet. *Queue*, 9(11):40:40–40:54, Nov. 2011.
- [4] O. Kerr. *Computer Crime Law*. Thomson/West, 2006.
- [5] D. Kravets. Wi-Fi-Hacking Neighbor From Hell Sentenced to 18 Years. *Wired Magazine* (Threat Level), July 2011.
- [6] M. Liberatore and et al. Forensic Investigation of Peer-to-Peer File Sharing Networks. In *Proc. DFRWS*, August 2010.
- [7] Microsoft.com. Sending small data segments over tcp with winsock.
- [8] R. P. Mislan, E. Casey, and G. C. Kessler. The growing need for on-scene triage of mobile devices. *Digital Investigation*, 6(3-4):112–124, 2010.
- [9] R. Shore. Pedophiles exploiting wireless loopholes. The Vancouver Sun, <http://www.canada.com/vancouver/news/story.html?id=cff3073b-ceea-4ba4-877f-d020715358e9>, February 13 2007.
- [10] D. Sicker, P. Ohm, and D. Grunwald. Legal issues surrounding monitoring during network research. In *Proc. ACM IMC*, pages 141–148, Oct. 2007.
- [11] U.S. General Accounting Office. File-Sharing Programs – Child Pornography Is Readily Accessible over Peer-to-Peer Networks. GAO-03-537T. Statement Before Congress of Linda D. Koontz, March 2003.
- [12] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley. Identifying 802.11 Traffic from Passive Measurements Using Iterative Bayesian Inference. In *Proc. IEEE INFOCOM 2006*.
- [13] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley. Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs. In *Proc. ACM IMC 2007*.
- [14] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley. Classification of access network types. In *Proc. IEEE INFOCOM 2005*.
- [15] S. Yang, J. Kurose, and B. Levine. Comprehensive study on disambiguation of residential wired and wireless access in a forensic setting. *Tech. Rep. UM-CS-2013-001, U. of Massachusetts Amherst*.