NAT usage in Residential Broadband Networks

Gregor Maier[1,3], Fabian Schneider[2,3], and Anja Feldmann[3]

[1] International Computer Science Institute, Berkeley, CA, USA
[2] UPMC Sorbonne Universités and CNRS, LIP6, Paris, France
[3] TU Berlin / Deutsche Telekom Laboratories, Berlin, Germany

Abstract. Many Internet customers use network address translation (NAT) when connecting to the Internet. To understand the extend of NAT usage and its implications, we explore NAT usage in residential broadband networks based on observations from more than 20,000 DSL lines. We present a unique approach for detecting the presence of NAT and for estimating the number of hosts connected behind a NAT gateway using IP TTLs and HTTP user-agent strings. Furthermore, we study when each of the multiple hosts behind a single NAT gateway is active. This enables us to detect simultaneous use. In addition, we evaluate the accuracy of NAT analysis techniques when fewer information is available.

We find that more than 90 % of DSL lines use NAT gateways to connect to the Internet and that 10 % of DSL lines have multiple hosts that are active at the same time. Overall, up to 52 % of lines have multiple hosts. Our findings point out that using IPs as host identifiers may introduce substantial errors and therefore should be used with caution.

1 Introduction

Today, network address translation (NAT) is commonly used when residential users connect their computers and laptops to the Internet. Indeed, most ISPs typically offer WiFi-enabled NAT home gateways to their broadband customers. These NAT gateways enable customers to easily and swiftly connect several devices to the Internet while needing only one public IP address. The prevalence of NAT devices and the number of terminals connected through a NAT gateway thus has implications on whether a public IP address can be used as a unique host identifier and if it is possible to estimate population sizes, e.g., malware infections, using IP addresses.

We, in this paper, analyze residential NAT usage based on anonymized packet-level traces covering more than 20,000 DSL lines from a major European ISP. We examine the number of DSL lines using NAT and how many distinct devices or hosts are connected via such NAT gateways. Furthermore, for DSL lines showing evidence of activity by more than one host we also study if these hosts are used concurrently.

While common wisdom holds that NAT is widely used in residential networks and that IP addresses are problematic end-host identifiers, no recent study reported numbers on NAT penetration or quantified the error potential in IP–to–end-host mappings. Most previous studies on identifying NAT gateways and inferring the number of hosts behind such gateways rely on information available in the packet headers, e. g., IPIDs, IP TTLs, or ports. Our approach takes advantage of HTTP user-agent information in addition to IP TTLs. In 2002, Bellovin [2] proposed and discussed the possibility to identify end-hosts by leveraging the fact that IPIDs are usually implemented as a simple counter. He found that this approach is limited in its applicability. Nowadays some IP-stacks even implement random IPIDs, further reducing the applicability of this approach. Beverly [3] evaluated several techniques to perform TCP/IP fingerprinting and found a host count inflation due to NAT by 9 % based on a one hour trace from 2004. Phaal [10] also takes advantage of the IP TTL. Furthermore, there is work in the area of OS fingerprinting, e. g., Miller [7].

Armitage [1] performed a measurement study in 2002 by offering Quake III servers at well connected Internet sites and monitoring the incoming connections. He identified NATed players by checking for non-default Quake client ports and found that 17–25 % of the players where located behind a NAT. Xie et al. [11] track IP-to-host bindings over time for counting hosts. However, they consider all hosts behind a NAT gateway as a single host. Casado et al. [4] use active web content to analyze NAT usage and IP address churn. By comparing local to public IP addresses they find that 5–10 % of IPs contacting *the monitored web services* have multiple hosts over a 7 month period.

In previous work [5] we showed that many distinct IP addresses are assigned to the same DSL line and that IP addresses cannot be used to reliably identify end hosts. While Casado et al. [4] found relatively low IP address churn, Xie et al. [12] came to a similar conclusion as we. In this paper we show that the situation is even worse because multiple hosts share one of these fluctuating IP addresses using NAT.

Our analysis of NAT usage shows that roughly 90 % of the studied lines connect to the Internet via a NAT gateway, presenting a high potential for IP ambiguity. Indeed, in our 24 h data sets 30–52 % of the DSL lines host multiple end-hosts. When considering shorter observation periods, 20 % of the DSL lines show activity from two or more hosts at least once within 1 hour. Even with time-frames as short as 1 sec, 10 % of the DSL lines show activity from multiple hosts. These results emphasize the large error potential of techniques that rely on an IP address to uniquely identify an end-host.

The remainder of this paper is structured as follows: We describe our data sets in Section 2 and explain our methodology in Section 3. Next, we present our results on NAT usage and the number of hosts in Section 4 and the impact of shorter time-scales in Section 5. We then critically discuss our findings in Section 6 and conclude in Section 7.

2 Data Sets

We base our study on multiple sets of anonymized packet-level observations of residential DSL connections collected at a large European ISP. Data anonymization and classification is performed immediately on the secured measurement infrastructure. Overall, the ISP has roughly 11.5 million (4%) of the 283 million worldwide broadband subscribers [8]. They predominantly use DSL. The monitor, using Endace monitoring cards, operates at the broadband access router connecting customers to the ISP's backbone. Our vantage point allows us to observe more than 20,000 DSL lines. The anonymized packet-level traces are annotated with anonymized DSL line card port-IDs. This enables us to uniquely distinguish DSL lines since IP addresses are subject to churn and as such cannot be used to identify DSL lines [5]. While we typically do not

Table 1. Overview of anonymized packet traces.

Name	Start date	Duration	Size
SEP08	Thu, 18 Sep 2008	24 h	≈ 4 TB
APR09	Wed, 01 Apr 2009	24 h	≈ 4 TB
AUG09a	Fri, 21 Aug 2009	24 h	≈ 6 TB
AUG09b	Sat, 22 Aug 2009	24 h	≈ 5 TB
MAR10	Thu, 04 Mar 2010	24 h	≈ 6 TB

experience any packet loss, there are several multi-second periods with no packets (less than 5 minutes overall per trace) due to OS/file-system interactions. Table 1 summarizes characteristics of the traces we used for our analysis, including the trace start, duration, and size.

3 Methodology

To analyze NAT usage among residential customers we have to *(i)* identify lines that use a NAT gateway (e. g., a home router) to connect to the Internet and *(ii)* differentiate between the hosts behind the NAT gateway.

3.1 Detecting the presence of NAT

To detect whether NAT is used on a DSL line, we utilize the fact that OSes networking stacks use well-defined initial IP TTL values (ttl_{init}) in outgoing packets (e. g., Windows uses 128, MacOS uses 64). Furthermore, we know that our monitoring point is at a well defined hop distance (one IP-level hop) from the customers' equipment. Since NAT devices do routing they decrement the TTLs for each packet that passes through them. We note that some NAT implementations might not decrement the TTL, however, per Section 6, we do not find evidence that such gateways are used by our user population in significant numbers.

These observations enable us to infer the presence of NAT based on the TTL values of packets sent by customers. If the TTL is $ttl_{init} - 1$ the sending host is directly connected to the Internet (as the monitoring point is one hop away from the customer). If the TTL is $ttl_{init} - 2$ then there is a routing device (i. e., a NAT gateway) in the customers' premises.

We note that users could reconfigure their systems to use a different TTL. However, we do not expect this to happen often. Indeed, we do find that almost all observed TTLs are between $ttl_{init} - 1$ and $ttl_{init} - 3$. While there are some packets with TTL values outside these ranges, they contribute less than 1.9 % of packets (1.7 % of bytes). Moreover, approximately half of those are due to IPSEC which uses a TTL of 255 and no other TTL has more than 0.44 % of packets. Given the low number of such packets, we discard them for our NAT detection.

A NAT gateway can come in one of two ways. It can be a dedicated gateway (e. g., a home-router) or it can be a regular desktop or notebook, that has Internet connection

Table 2. First network activity example

From Pkt Hdr		From HTTP User-Agent		
TTL	Proto	OS	Family	Version
63	53/DNS	–	–	–
126	80/HTTP	Win2k	Firefox	2.0.1
126	80/HTTP	WinXP	Firefox	3.0.2
126	80/HTTP	WinXP	MSIE	6
126	80/HTTP	WinXP	Firefox	2.5.1

Table 3. Second network activity example

From Pkt Hdr		From HTTP User-Agent		
TTL	Proto	OS	Family	Version
63	53/DNS	–	–	–
63	80/HTTP	Linux	Firefox	3.0.1
62	80/HTTP	Linux	Firefox	3.0.1
126	80/HTTP	WinVista	MSIE	8
126	80/HTTP	WinVista	Firefox	3.0.2

sharing activated. A dedicated NAT gateway will often directly interact with Internet services, e. g., by serving as DNS resolver for the local network or for synchronizing its time with NTP servers. Moreover, they generally do not surf the Web or use HTTP.

3.2 Number of hosts per DSL line

We also want to count how many hosts are connected to each DSL line behind a NAT gateway to enable us to estimate the ambiguity when using IP addresses as host identifiers. A first step towards identifying a lower bound for the number of hosts per line is to count the number of distinct TTLs observed per line. Recall that Windows uses a ttl_{init} of 128 and that MacOS X and Linux use 64 and that most of the observed TTL values are within the ranges of 61–63, and 125–127. These ranges are far enough apart to clearly distinguish between them at our monitoring point. Therefore, we can use observed TTLs to distinguish between Windows and non-Windows OSes, yet we cannot distinguish between distinct Windows systems. This is unfortunate, as analyzing HTTP user-agents shows that Windows is the dominant OS in our user population.

However, we can use additional information to distinguish hosts. HTTP user-agent strings of regular browsers (as opposed to user-agent strings used e. g., by software update tools or media players) include information about the OS, browser versions, etc. This can help us differentiate between hosts within the same OS family. We find that up to 90 % of all active DSL lines have user-agent strings that contain such OS and browser version information. In addition, we often observe several different OS and browser combinations on a single line. We theorize, that home-users tend to keep preinstalled (OS and browser) software, rather than installing the same software on each of their machines.

For example, consider the summary of all network activity of one DSL line in Table 2. We see a directly connected device (TTL 63 == $ttl_{init} - 1$) that is only using DNS. According to our definition in Section 3.1 this device is classified as a dedicated NAT gateway. We also observe TTLs of 126, which is consistent with a Windows OS behind a NAT gateway. Examining the HTTP user-agent strings we see that both Win2k and WinXP are present. Thus, we can assume that there are at least two distinct hosts behind the NAT gateway. However, we also see that the WinXP OS uses several different browser families and versions. While it can happen that users use two different browser families on a single host (e. g., MSIE and Firefox), it seems rather unlikely that they use different *versions* of the same browser family on the same host. Using this rationale, the two different Firefox versions on WinXP indicate two distinct hosts, yielding a total of 3 end-hosts.

Or consider the example in Table 3. Here we also see a directly connected device (TTL 63), however there is also HTTP activity with the same TTL. We therefore classify this device as a host. We also see TTLs that are consistent with NATed Windows and Linux systems, so we conclude that the directly connected device serves a dual function: as NAT gateway and as regular computer. Moreover, we see one OS/browser combination with TTL 62—another host. For TTL 126 we see only WinVista as OS but two different browser families, which likely indicates just one host with both Firefox and MSIE installed. Overall, we infer for this example that there are 3 active hosts.

3.3 A NAT analysis tool

We develop a small C program, `ttlstats`[1], to implement our NAT analysis. For each DSL line, the tool records whether a particular protocol was used by that line, which TTL was used in packets of this protocol, and for HTTP which user-agents were used. To identify protocols we use their well-known ports, which works well for the protocols we consider [5].

For HTTP we parse the user-agent strings and extract the operating system (OS) version and the browser version. We limit our analysis to user-agent strings from typical browsers (Firefox, Internet Explorer, Safari, and Opera), user-agents from mobile hand-held devices (see [6]), and gaming consoles (Wii, Xbox, PlayStation). We do not consider other user-agents (e. g., from software update clients) since those often do not include OS information or host identifiers. To estimate a lower bound for the number of hosts behind a NAT gateway we use two approaches:

OS only We only count different $\langle$TTL,OS$\rangle$ combinations as distinct hosts.

OS + browser version For each $\langle$TTL,OS$\rangle$ combination we also count the number of different browser versions from the same browser family as distinct hosts. Firefox and Internet Explorer are examples of browser families. We do not consider different browser families as additional hosts.

In our first example above, OS only yields a host count of 2 while OS + browser version yields a host count of 3. In our second example both counting methods yield a host count of 3: one Linux system that is used as gateway and regular computer, one NATed Linux system, and one computer with Windows Vista.

3.4 NAT analysis for different data set types

Often the kind of data (anonymized packet-level information with HTTP) we use for this NAT analysis is not available. However (anonymized) HTTP logs might be more readily available. Yet, IP/TCP header only traces are common in the measurement community as well. Thus, we compare how well NAT analysis schemes perform when less information is available. For this we use several reduced information data sets, and repeat the analysis.

[1] Our analysis scripts available online.

Table 4. Overview of results. Top three rows are relative to total number of active lines, remaining rows are relative to “Lines with active hosts” (B.2), i. e., for C.1–E.2 100 % is equivalent to B.2.

Ref.	Description	SEP08	APR09	AUG09a	AUG09b	MAR10
A.1	Lines using NAT	89 %	91 %	92 %	92 %	93 %
B.1	Lines on which only dedicated NAT is active	9 %	10 %	14 %	18 %	10 %
B.2	Lines with active hosts (NATed and unNATed)	91 %	90 %	86 %	82 %	90 %
C.1	Lines with unNATed Windows	9 %	8 %	7 %	7 %	6 %
C.2	Lines with unNATed Linux/Mac	1 %	1 %	1 %	1 %	1 %
D.1	Total systems (OS only)	141 %	142 %	143 %	140 %	145 %
D.2	Total systems (OS + browser version)	155 %	162 %	179 %	172 %	185 %
E.1	Lines with $>$ 1 host (OS only)	30 %	31 %	31 %	30 %	32 %
E.2	Lines with $>$ 1 host (OS + browser version)	36 %	39 %	49 %	46 %	52 %

4 NAT usage/hosts per DSL line

In this section we present the results from our NAT analysis. We first discuss the prevalence of NAT devices at DSL lines before continuing with the number of hosts per line. Finally, we investigate NAT detection with different data set types.

4.1 NAT usage

Overall, we find that NAT is prevalent and that the vast majority of DSL lines use NAT to connect hosts to the Internet. We also find that a significant number of lines connects more than one host. Table 4 summarizes our key findings. Note that we term a device or host *active* if it sent IP packets during the trace. More than 90 % of lines utilize NAT (Table 4, row A.1). This result differs from the findings of Armitage [1] from 2002 who only found 25 % of the IPs were behind a NAT. On 9–18 % of lines (B.1) we only observe traffic that we attribute to the NAT gateway and no traffic from regular hosts[2]. We note that this traffic could also be caused by a directly connected, unused host. However, unused hosts might still check for software or anti-virus updates using HTTP, and would thus be counted as a host. The remaining lines (82–91 %, B.2) have active hosts (those lines may or may not be NATed).

We next take a closer look at DSL lines with active hosts and determine how many of these lines are using NAT. We find that only 7–10 % (C.1 and C.2) of lines with active hosts are not NATed, i. e., there is only one host which is directly connected.

Finally, we investigate how many more hosts than lines are present: the ratio of detected hosts to the number of lines. In rows D.1 and D.2 we show the number of observed hosts relative to the number of lines with active hosts. For D.1 we use the heuristic which counts every unique TTL and OS combination as a separate host (OS only). For row D.2 we also increment the per line host count if we observe TTL-OS combinations with multiple versions of the same browser family (OS + browser version). According to our definition, we will always see more hosts than lines with active hosts.

[2] I. e., we observe only traffic with TTL 63 and no HTTP activity.

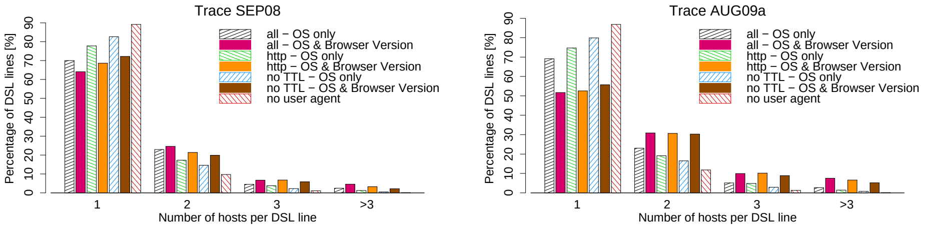

Fig. 1. Fraction of DSL lines vs. number of hosts per line for SEP08 and AUG09a

However, the differences are strikingly large—up to 1.85 times as many hosts than lines in MAR10 using the OS + browser version counting method. Independent of the estimation method the number of hosts behind NAT devices, our host counts, are far larger than the estimations by Beverly [3] from 2004, who estimated 1.09 times more hosts than IPs. This difference might be due to 6 additional years of NAT gateway deployment, different vantage points (Internet peering/exchange point vs. broadband access), different observation periods (1 h vs. 24 h), and/or information base (SYN trace vs. TTL plus HTTP logs).

4.2 Number of hosts per line

Given that we see so many more hosts than lines with active hosts, we next investigate lower bounds for the number of lines with more than one host. A large fraction of such lines implies many public IP addresses with more than one host, thus limiting the utility of IPs as host identifiers. We see that 30–52 % of lines have more than one active host (Table 4, rows E.1 and E.2). We note that between APR09 and AUG09a the number of lines with more than one host increases significantly (OS + browser version, row E.2). We attribute this to an increase in browser heterogeneity: Following the release of MSIE 8 in late March 2009, we observe a significant share of MSIE 6, 7, and 8 in AUG09, while only MSIE 6 and 7 have a significant share in SEP08 and APR09. Consider the example that two hosts use a DSL-line and both have WinXP and MSIE 7. In this case we cannot distinguish between them. However, if one is upgraded to MSIE 8 while the other is not, then we can distinguish them.

In Figure 1 we present a more detailed look by plotting the fraction of lines with *n* hosts. We only present plots for SEP08 and AUG09a, the other traces exhibit similar behavior. We focus on the bars labeled “all” first. Note that we observe up to 7 % of lines with more than 3 hosts. We also investigate whether this high number of lines with multiple hosts is due to several computers (PCs or Macs) that are used via the same line or whether mobile hand-held devices (e. g., iPhones), or game consoles (e. g., Wii) are responsible for this. We identify these devices by examining the HTTP user-agent string. If we exclude mobile hand-held devices and game consoles, still 25–28 % (OS only; 34–45 % with OS + browser version) of lines have more than one host (not shown). Therefore, we conclude that the number of DSL lines with multiple end-hosts is only slightly influenced by mobile devices. In [6], we investigated mobile device usage in detail.

4.3 NAT analysis with different data set types

As discussed in Section 3.4, we also use reduced data sets ("http", "no TTL", and "no useragent") and compare the NAT usage estimates to those based on the full data set available to us ("all"). Figure 1 compares the number of hosts per line for the different data sets. Note, without HTTP user-agent data there is no difference between the scheme for OS only and OS + browser version. Most accuracy is lost when relying on IP TTL only ("no useragent"). Removing the IP TTL ("no TTL") information shows slightly better results. Compared to "all" information using HTTP logs annotated with TTL information (but discarding all non-HTTP activity, "http") gives a very good estimate of NAT prevalence.

5 Impact of shorter time-scales

So far we have limited our discussion to a static view of NAT behavior, i. e., we analyzed whether a DSL line is NATed and how many hosts are connected via this line. If a line has more than one host, IP addresses cannot be reliably used as host identifiers when considering time-scales of one day (our trace duration). However, it is possible that even though a line has two hosts, the first host is only active in the morning while the second host is only active in the evening. Thus, although the line has two hosts, they are not used at the same time. This can reduce the ambiguity of using IP addresses as host identifiers over smaller time intervals (e. g., by utilizing timeouts).

5.1 Analysis approach

To answer if multiple devices are used at the same time, we compute the *minimal* inter activity time (*m*IAT) between any two HTTP requests issued by two different host on the same DSL line. If we observe an *m*IAT of T seconds then we know that two or more distinct hosts were active at this line within T seconds. As we need timestamps for this analysis we cannot use the output of the `ttlstats` tool (Section 3.3) as it aggregates all activity of a line for scalability reasons. Therefore, we revert to using HTTP request logs, which corresponds to the "http" data type and use the OS only counting method. These logs include timestamps for every request. We rely on Bro [9] for HTTP parsing.

5.2 Results

In Figure 2 we plot the fraction of lines with two or more hosts for increasing *m*IATs. This plot enables us to study how close in time two (or more) hosts are active via the same line. This allows us to estimate by how much ambiguity can be reduced by using a timeout, i. e., by using the IP-to-host mapping only for a limited time.

Even with intervals as low as 1 sec we observe more than 10 % of DSL lines with multiple hosts (12 % for MAR10). When considering *m*IATs of 1 h, around 20 % of lines have activity from multiple hosts (18 % for SEP08 up to 22 % for MAR10). We thus conclude that if a line has multiple hosts they are likely active at the same time or within a short time period. We see the lines starting to level off at around 10 h. This

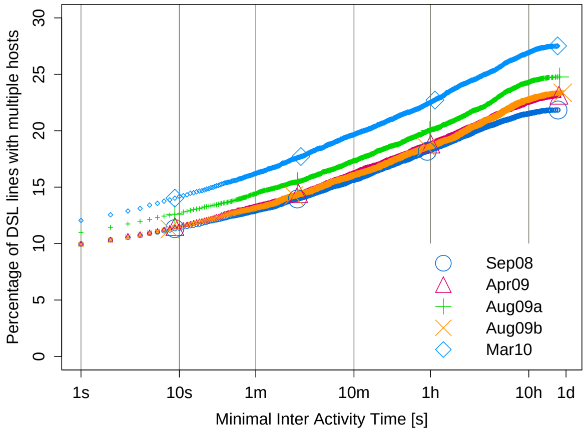

Fig. 2. Fraction of DSL lines with more than one active host within a particular time interval using OS only.

is likely due to the time interval that users actively use their computers, as opposed to using them around the clock. We confirm these results by applying the static analysis (see Section 3.2 and Section 4.2) for slices of the traces, i. e., we subdivide each trace into time bins of 1, 5, 10, 30, and 60 minutes and repeat the analysis for each bin.

6 Discussion

This study aims at estimating the number of active end-host per DSL line. Our methodology will likely underestimate the number of end hosts per lines, since we cannot distinguish between hosts with identical OS and browser software. This actually renders our approach problematic for networks with homogeneous OS/software installations (e. g., businesses). However, our approach already reveals a significant number of hosts per DSL line. Furthermore, the DSL lines in our data sets are for residential customers. The ISP also offers different but comparable DSL plans for small businesses. Parsing additional application protocol headers might reveal additional hosts that were not counted, e. g., P2P peer IDs, however only a small fraction of DSL lines use P2P [5].

On the other hand there are factors that can bias our results towards overestimating the number of hosts per DSL line: Our method counts a computer that has two OSes installed (e. g., in a dual-boot or virtualized setup) as two different hosts. Yet, it is questionable if it is wrong to count them as separate hosts. Likewise, if a user updates his browser during our observation period we also count the same machine twice. However, these artifacts decrease as we consider shorter time-frames since it requires time to reboot another OS and/or update a browser. Therefore, the results for small *m*IATs are reasonable lower bounds for the number of hosts per line.

We further note that some NAT gateway might not decrement the TTL. If such a NAT gateway is used, we would classify the DSL as unNATed. However, if *multiple* hosts are connected through such a gateway, we are able to detect them. We have not found any evidence that a significant number of such non-decrementing gateways is used by our user population.

7 Conclusion

We presented a novel approach for detecting DSL lines that use network address translation (NAT) to connect to the Internet. Our approach is able to infer the presence of a NAT device and to provide lower bounds for the number of hosts connected behind the NAT gateway. For lines with multiple hosts connected we also studied the temporal behavior to see whether multiple hosts are active at the same time. Our approach relies on IP TTL information and HTTP user-agent strings and we analyze the accuracy when using less information (e. g., TTLs only, or user-agent strings only) for the NAT analysis. We find that most accuracy is lost when user-agent strings are omitted.

We find that 10 % of DSL lines have more than one host active *at the same time* and that 20 % of lines have multiple hosts that are active within one hour of each other. Overall 30–52 % of lines have multiple hosts. These results underscore the perils involved when using IPs as host identifiers.

In future work we plan to investigate NAT behavior over a number of consecutive days and to augment our analysis with IPIDs and ephemeral ports. Combining IP address churn [5] and NAT behavior, we further plan to assess the effect and potential error of utilizing IPs as host identifiers.

References

1. Armitage, G. J. Inferring the extent of network address port translation at public/private internet boundaries. Tech. Rep. 020712A, Center for Advanced Internet Architectures, 2002.
2. Bellovin, S. M. A technique for counting natted hosts. In *Proc. Internet Measurement Workshop (IMW)* (2002).
3. Beverly, R. A robust classifier for passive TCP/IP fingerprinting. In *Proc. Conference on Passive and Active Measurement (PAM)* (2004).
4. Casado, M., and Freedman, M. J. Peering through the shroud: The effect of edge opacity on ip-based client identification. In *Proc. USENIX NSDI* (2007).
5. Maier, G., Feldmann, A., Paxson, V., and Allman, M. On dominant characteristics of residential broadband internet traffic. In *Proc. Internet Measurement Conference (IMC)* (2009).
6. Maier, G., Schneider, F., and Feldmann, A. A first look at mobile hand-held device traffic. In *Proc. Conference on Passive and Active Measurement (PAM)* (2010).
7. Miller, T. Passive OS fingerprinting: Details and techniques. `http://www.ouah.org/incosfingerp.htm` (last modified: 2005).
8. OECD. Broadband Portal. `http://www.oecd.org/sti/ict/broadband`, Dec. 2009.
9. Paxson, V. Bro: A system for detecting network intruders in real-time. *Computer Networks Journal 31*, 23–24 (1999). Bro homepage: `www.bro-ids.org`.
10. Phaal, P. Detecting NAT devices using sFlow. `http://www.sflow.org/detectNAT/` (last modified: 2009).
11. Xie, Y., Yu, F., and Abadi, M. De-anonymizing the internet using unreliable ids. In *Proc. ACM SIGCOMM Conference* (2009).
12. Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., and Wobber, T. How dynamic are IP addresses? In *Proc. ACM SIGCOMM Conference* (2007).

References

1. Armitage, G. J. Inferring the extent of network address port translation at public/private internet boundaries. Tech. Rep. 020712A, Center for Advanced Internet Architectures, 2002.
2. Bellovin, S. M. A technique for counting natted hosts. In *Proc. Internet Measurement Workshop (IMW)* (2002).
3. Beverly, R. A robust classifier for passive TCP/IP fingerprinting. In *Proc. Conference on Passive and Active Measurement (PAM)* (2004).
4. Casado, M., and Freedman, M. J. Peering through the shroud: The effect of edge opacity on ip-based client identification. In *Proc. USENIX NSDI* (2007).
5. Maier, G., Feldmann, A., Paxson, V., and Allman, M. On dominant characteristics of residential broadband internet traffic. In *Proc. Internet Measurement Conference (IMC)* (2009).
6. Maier, G., Schneider, F., and Feldmann, A. A first look at mobile hand-held device traffic. In *Proc. Conference on Passive and Active Measurement (PAM)* (2010).
7. Miller, T. Passive OS fingerprinting: Details and techniques. `http://www.ouah.org/incosfingerp.htm` (last modified: 2005).
8. OECD. Broadband Portal. `http://www.oecd.org/sti/ict/broadband`, Dec. 2009.
9. Paxson, V. Bro: A system for detecting network intruders in real-time. *Computer Networks Journal 31*, 23–24 (1999). Bro homepage: `www.bro-ids.org`.
10. Phaal, P. Detecting NAT devices using sFlow. `http://www.sflow.org/detectNAT/` (last modified: 2009).
11. Xie, Y., Yu, F., and Abadi, M. De-anonymizing the internet using unreliable ids. In *Proc. ACM SIGCOMM Conference* (2009).
12. Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., and Wobber, T. How dynamic are IP addresses? In *Proc. ACM SIGCOMM Conference* (2007).