# Exploring Visible Internet Hosts through Census and Survey

## USC/ISI Technical Report ISI-TR-2007-640

John Heidemann[1,2], Yuri Pradkin[1], Ramesh Govindan[2], Christos Papadopoulos[3], Joseph Bannister[4]
[1] USC/Information Sciences Institute [2] USC/Computer Science Dept.
[3] Colorado State University [4] The Aerospace Corporation

## Abstract

Measurement studies published in the literature have, for the most part, ignored the population of hosts. While many hosts are hidden behind firewalls and in private address space, there is much to be learned from examining the population of *visible* Internet hosts—one can better understand network growth and accessibility and this understanding can help to assess vulnerabilities, deployment of new technologies, and improve network models.

This paper is, to our knowledge, the first attempt to measure the population of visible Internet edge hosts. We measure hosts in two ways: via periodic *Internet censuses*, where we query *all* accessible Internet addresses every few months, and via *surveys* of a small fraction of the responsive address space, probing each address every 11 minutes for one week. These approaches are complementary: a census is effective at evaluating the Internet as a whole, while surveys validate the census and allow observation of the lifetime of typical address occupancy.

We find that only 3.6% of allocated addresses are actually occupied by visible hosts, and that occupancy is unevenly distributed, with a quarter of responsive /24 subnets less than 5% full, and only 9% of subnets more than half full. We establish an upper-bound on the number of servers in the Internet at 36 million, about 16% of the responsive addresses. Many firewalls are visible and we observe significant diversity in the distribution of firewalled block size. While the absolute number of firewalled blocks appears stable, the ratio of coverage of visible firewalls to the number of visible addresses is declining, perhaps suggesting increasing use of invisible firewalls.

## 1 Introduction

Measurement studies of the Internet have focused primarily on network traffic and the network topology. Many surveys have characterized network traffic in general and in specific cases [21, 28, 9, 36, 15]. More recently, researchers have investigated network topology, considering how networks and ISPs connect at the AS [11, 39, 13, 24, 8] and router level [40, 22]. These studies have yielded insight into how traffic in the network, business relationships, routing opportunities and risks, and network topology.

For the most part these studies have ignored the population of hosts at the *edge* of the network. Yet there is much to be learned from understanding end-host characteristics. Today, many simple questions about hosts are unanswered: How big is the Internet, in numbers of hosts? How densely do hosts populate the IPv4 address space? How many hosts are, or could be, clients or servers? How many hosts are firewalled or behind address translators?

While simple, these questions have profound implications for network and protocol design. Today peer-to-peer technologies and voice-over-IP imply that many end-hosts wish to receive connections, yet widespread use of NAT and firewalls prohibit unsolicited inbound connections. Can we begin to quantify the prevalence of some of these middleboxes? Widespread use of laptops and dynamically addressed last-mile approaches such as cable modems and DSL connections mean that many hosts today no longer have stable addresses. How long is a dynamic address used by one host? Dynamic address stability influences reputation systems, peer-to-peer client churn, and potential vulnerability of dynamic hosts to malware. Understanding these questions today is important, but more important still is understanding their trends over time. What is the rate of address allocation, the trend in host inaccessibility, or the change in host attachment duration? In this paper, we take some initial steps towards answering some of these questions.

Many hosts today are unreachable, hidden behind network-address translators and firewalls. Yet much can be learned by studying the *visible* Internet population of end hosts. We define the visible Internet as the set of hosts on the globally routable address space that respond to queries. As we shall see later, this definition provides indirect evidence about Internet hosts hidden behind firewalls, NATs, and private addresses. To our knowledge, this paper represents the first attempt to study this population directly.

This paper makes three contributions. The first is to develop two new methodologies to measure Internet end-hosts. We conduct *Internet censuses* where we query all visible Internet addresses to gain a comprehensive but infrequent estimate of the end-host population, probing each address every three months (Section 2). We also take *surveys* of about 1% of the responsive /24 subnets to gain a detailed estimate of a representative fraction of the end-host population at a much finer timescale, sampling each address every 11 minutes (Section 3). We have taken multiple censuses and surveys to allow us to study trends in the population. We are not the first to actively probe the Internet—viruses such as Code Red engage in massively parallel probing, and tools like Mercator [14] and Rocketfuel [38] discover router-level topology—but we are the first to use controlled, probing of all visible addresses to understand end-host behavior. Census and survey bring trade-offs in temporal and spatial sampling that we explore in Section 4.

Second, we evaluate this data to estimate the number of *populated addresses* in the Internet (Section 2.4). We find that only 3.6% of allocated addresses are actually occupied by visible hosts, and that occupancy is unevenly distributed, with a quarter of responsive /24 subnets less than 5% full, and only 9% of subnets more than half full. We also define the number of *stably populated addresses*, that is, the number of addresses that consistently respond to queries (Section 5). We believe that the number of stably populated addresses is an upper bound on the number of hosts that can function as traditional Internet servers, and we find that number to be 36 million, or about 16% of the responsive addresses. Finally, we estimate the *lifetime of intermittently populated addresses* (Section 3.3). Because dynamically assigned addresses can be used by many different hosts, we cannot estimate the client population, but typical attachment durations represent an important characteristic of client host behavior; we show that 50% addresses are continuously occupied for 81 minutes or less.

Finally, we evaluate this data to estimate *trends in the deployment of firewalls* on the public Internet (Section 6). Firewalls respond to probes in several different ways, perhaps responding negatively, or not responding at all, or in some cases varying their response over time [35, 4]. Estimating the exact number of firewalls is therefore quite difficult. However, we present trends in firewalls that respond negatively over seven censuses spread over 15 months. Many such firewalls are visible and we observe significant diversity in the distribution of firewalled block size. While the absolute number of firewalled blocks appears stable, the ratio of coverage of visible firewalls to the number of visible addresses is declining, perhaps suggesting increasing use of invisible firewalls.

We explore the limitations of active probing (Section 7), including non-responsive, multi-homed hosts, and probe loss. We validate our key results with measurements taken from multiple locations on the Internet (Section 8.2), and at different times. To support external validation of our results and additional analysis using our methodology, we have made both the datasets described here [30, 41] and the software tools [29] we used to collect the data publicly available.

Our experience with censuses and surveys suggests that both have their place in studies of Internet addresses. Periodic censuses are important to give perspective on the whole Internet; we use them to seed surveys, for example (Section 3.1). Furthermore, censuses and surveys can be used to validate each other; we use census data at suitable levels of aggregation to validate availability (Section 8.1). We find that surveys are best at capturing phenomena that vary at relatively fine timescales, such as host availability. On the other hand, censuses can better capture long-term trends of slowly-varying, but spatially non-uniform measures, such as the prevalence of firewalls.

## 2 A Census of the Internet Address Space

We first consider taking a *census* of the Internet. By definition, a census attempts to enumerate all possible members of a population. While this approach might seem ideal, with a large and changing population it can actually be less accurate than a well-conducted survey of a fraction of the population. We compare our census to survey results in Section 8.1.

Our motivation for conducting a census of the Internet address space was to characterize its diversity. Since we began with no understanding of the variation in address usage across the Internet, this approach seemed best able to capture whatever range was present.

There are several challenges to conducting an Internet census. At first glance, the large number of addresses seems daunting, but there are only $2^{32}$, and only about half of these are routable today, so a relatively modest probe rate of 1000 probes/s (about 256kb/s) can enumerate the entire space in 49 days. The primary challenges to probing are to probe in a manner that is unlikely to be confused with malicious scans, and to understand the effects of lost probes on the results. We describe our probe implementation below (Section 2.2).

We have completed 15 censuses with this approach as shown in Table 1. We show preliminary results of the Internet address space below (Section 2.4), then use this data later to investigate Internet firewall trends (Section 6).

### 2.1 Census Design Issues

Our census must consider who is probed, what kind of probe is used, what replies are recorded, and how probes are managed. We consider each of those issues next.

**Who:** Our census is an enumeration of the allocated Internet address space at the time the census is conducted (ignoring the effects of transient hosts). We discard private address space [32] and multicast addresses. We also did not probe addresses with last octet 0 or 255 since those are often unused or allocated for local broadcast in /24 networks. We determine the currently assigned address space from IANA [2]. This list is actually a superset of the routable addresses, since addresses may be assigned to registrars but not yet injected into global routing tables [23].

In an ideal world a census would exactly capture the Internet at given moment in time, however, in practice, it takes some time to carry out a census, and the Internet changes over this time. However, we show that differences in concurrent censuses are relatively small and unbiased in Section 8.2.

**Requests:** For each address, we send a single probe message and then record the time until a reply is received as well any the (positive or negative) reply code. If no reply is received after a liberal timeout (currently 5s), we record that as well.

There are several possible choices for probe protocols, including TCP, UDP, and ICMP. Our choice was determined by our requirement for *response ubiquity:* all hosts must understand our probes and react in predictable way, sending back some kind of response that we could use as indication that the host is alive. A secondary requirement is that probes are non-threatening and not confused with denial-of-service attacks.

We selected ICMP (type 8, echo request) because it is relatively widely supported and generally considered benign. We used TCP in one early census ($TCP_1$), because it is blocked by fewer firewalls, but we reverted to ICMP echo requests after receiving significantly more complaints from network administrators. Comparing contemporaneous TCP and ICMP surveys gives the same order of magnitude hosts, so we think alternative protocols would not greatly change our results.

**Replies:** Each ICMP echo request can result in several potential replies [3] with the following interpretations:

**Positive acknowledgment:** We receive an *echo reply* (type 0), indicating positively the presence of a host at that address.

**Negative acknowledgment:** We receive an *destination unreachable* (type 3), indicating that host is either down or doesn't exist. In Section 6 we subdivide negative replies based on response code, interpreting codes for *network, host*, and *communication administratively prohibited* (codes 9, 10, and 13) as positive indication of a firewall.

**No reply:** Lack of response can have several possible causes. First, either our probe or subsequent response could have accidentally failed to reach the destination due to congestion or partitioning. Second, it may have intentionally failed to reach the destination due to firewall. Third, the host may not exist or be down and its last-hop router may decline to generate any ICMP reply.

Only reply types 0 and 3 are usually solicited by an echo request. We see a tiny fraction of other replies, which we classify as non-administrative negative acknowledgments.

**Request frequency:** To avoid appearing malicious, we probe in a pseudo-random sequence, so that the probes to any portion of the address space are dispersed in time. This approach also reduces the effect of correlated outage of portions of the address space. While outage of a block of addresses may affect some probes, probes to adjacent addresses are at different times, so the probe loss rate is effectively independent and proportional to the block outage time over the whole census duration.

One design issue we may reconsider is the use of duplicate probes for addresses that fail to respond. A second probe reduces the effects of probe loss, but it increases the cost of the census. Instead, we opted for more frequent censuses rather than a more reliable single census. We consider the effects of loss below in Section 7.3.

## 2.2 Census Implementation

We have implemented our census taker as a simple C++ program. It implements rate limiting by maintaining a fixed array of currently outstanding probe control blocks (PCBs). PCBs time out after a fixed controlled interval (5 seconds) and are replaced by newly constructed PCBs with different destination addresses. Thus, the effective probe rate is determined by the ratio of the PCB array size (currently 1200 entries) to the timeout. A scheduler spreads probes out within the time-out interval in order to reduce bursts.

We use the ICMP identifier field to match ICMP replies with PCBs. ICMP sequence numbers are used as indices in the PCB array for fast PCB lookups. A vast majority of replies are matched by this mechanism, but in some cases (remote ICMP implementations that do not echo the source message and sequence number) we resort to searching by IP address. We have also experimented with sending a 32-bit random cookie in the body of ICMP message to identify the probe, but many ICMP implementations do not return this cookie in the ICMP response.

Our census taker must enumerate the entire address space in an order that touches adjacent addresses at very different times. Our current implementation (in use since $IT_{11}$) uses a three-step algorithm. First, it enumerates all 32-bit addresses in order, guaranteeing completeness. To disperse probes to any given subnet across time, we bit-reverse this address, so that any $n$-bit block is probed only once every $2^{32-n}$ probes. Each trace also exclusive-ors the series with an arbitrary constant, ensuring that each trace uses a different absolute order. This algorithm can be checkpointed with only 64 bits of state, and it parallelizes easily (currently over four machines). Finally, we filter potential probe addresses through the list of allocated addresses using balanced binary trees (C++ STL red-black trees) for efficient search.

We repeat censuses about every three months. Since March 2006, each each census has been conducted concurrently from two sites, one on the east and west coasts of the United States. Each site uses four probing machines, all connected to a single Ethernet segment. The aggregate bandwidth required for our probes and responses is approximately 166kb/s. The Internet connection at the western site is well overprovisioned, but we consume about 30% of the Internet connection capacity at the east coast site.

## 2.3 Census Datasets

Table 1 shows all 16 censuses collected to date. We also observe one anomaly in these datasets. The number of NACKs in $IT_{11w}$ and $IT_{12w}$ was about a factor of ten higher than all other datasets. We believe this represents a temporary routing configuration anomaly in our network: about 90% of these NACKs are from a single backup router peering with commercial Internet ISP (Sprint) that was configured at the time to send ICMP-unreachables in response to unroutable packets. These extra NACKs do not affect our conclusions because they were ICMP type 3, code 1 (Destination Host Unreachable), and so are not considered in our analysis.

## 2.4 Preliminary Results

To demonstrate the power of an Internet census, we next present two brief studies that can be drawn from this data.

**Trends in Internet Address Allocation:** Since the IPv4 address space is finite and limited to 32 bits, the rate of address allocation is important. In fact, concerns about address space

|  |  | Dur. | Alloc. | ACKs | NACKs | (ad.proh. |
|---|---|---|---|---|---|---|
| **Name** | **Start Date** | (days) | ($\times 10^9$) | ($\times 10^6$) | ($\times 10^6$) | $\times 10^6$) |
| *ICMP₁* | 2003-06-01 | 117 | 2.52 | 51.08 | n/a | n/a |
| *ICMP₂* | 2003-10-08 | 191 | 2.52 | 51.52 | n/a | n/a |
| *TCP₁* | 2003-11-20 | 120 | 2.52 | 52.41 | n/a | n/a |
| $IT_1$ | 2004-06-21 | 70 | 2.40 | 57.49 | n/a | n/a |
| $IT_2$ | 2004-08-30 | 70 | 2.40 | 59.53 | n/a | n/a |
| $IT_4$ | 2005-01-05 | 42 | 2.43 | 63.15 | n/a | n/a |
| $IT_5$ | 2005-02-25 | 42 | 2.43 | 66.10 | n/a | n/a |
| $IT_6$ | 2005-07-01 | 47 | 2.65 | 69.89 | n/a | n/a |
| $IT_7$ | 2005-09-02 | 67 | 2.65 | 74.40 | 46.52 | 17.33 |
| $IT_9$ | 2005-12-14 | 31 | 2.65 | 73.88 | 49.04 | 15.81 |
| $IT_{11w}$ | 2006-03-07 | 24 | 2.70 | 95.76 | 740.44 | 17.84 |
| $IT_{12w}$ | 2006-04-13 | 24 | 2.70 | 96.80 | 723.82 | 16.94 |
| $IT_{13w}$ | 2006-06-16 | 32 | 2.70 | 101.54 | 77.11 | 17.86 |
| $IT_{14w}$ | 2006-09-14 | 32 | 2.75 | 101.17 | 51.17 | 16.40 |
| $IT_{15w}$ | 2006-11-08 | 62 | 2.82 | 102.96 | 84.44 | 14.73 |

Table 1: IPv4 address space allocation (alloc.) and responses over time (postive and negative acknowledgments, and NACKs that indicate administratve prohibited), Censuses before September 2005 did not record NACKs.

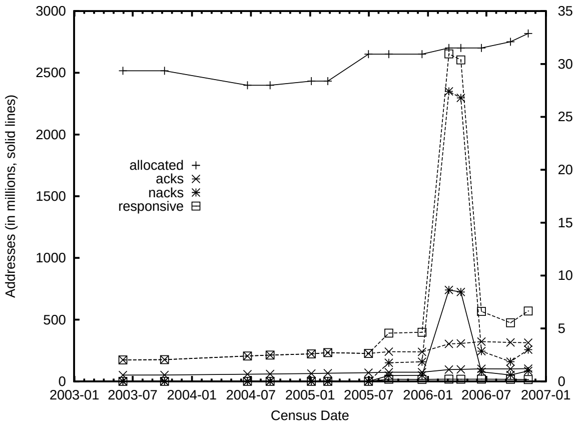


Figure 1: IPv4 address space allocation and utilization over time. Solid lines indicate absolute values, dashed are percentages of allocated addresses. (Data from all censuses.)

exhaustion [16] were the primary motivation for IPv6 [7] and CIDR [12] as an interim conservation strategy. They also motivated deployment of Network Address Translation (NAT) devices that allow many computers to share a single globally routable address [42]. We next consider how effective conservation of address space allocation has been 20 years after these initial studies.

Figure 1 and Table 1 show trends in address space allocation and utilization computed over the each individual Internet address. To put these values into context, a total of $2^{32}$ addresses are possible, but to date only only 2.8 billion have been allocated, and we can further eliminate, and we can further eliminate private and multicast address space. Finally, this evaluation represents the number of *addresses* and not actual host computers, since multiple computers may be hidden behind a single NAT box.

We can see that allocation is at about 106M addresses/year, and the visible hosts are growing at at 17.2M addresses/year.

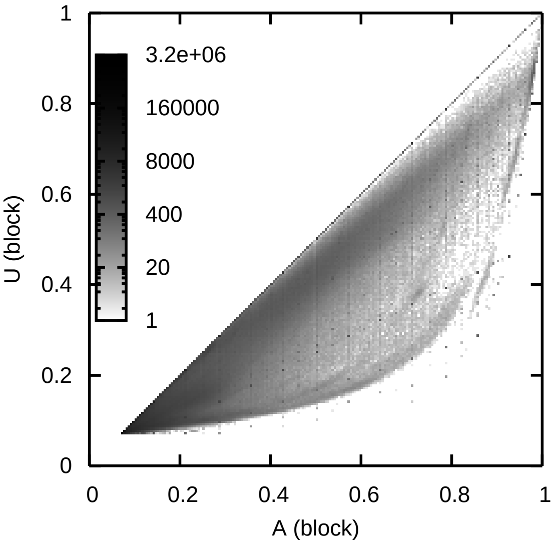


Figure 2: Density of all affirmative /24 Internet address blocks, grouped by block availability and block uptime (data from all 15 censuses).

This evaluation is somewhat can be difficult to interpret, though, because address allocation is far from uniform. Many ISPs give out individual addresses to users, but these addresses are usually dynamic and change over time. Even users of “always-on” connections may shift addresses over time. Businesses and ISPs, on the other hand, are given addresses in power-of-two blocks, which are rarely filled.

**Characterizing Internet Address Blocks:** To begin to characterize the Internet, Figure 2 shows a density plot of *availability* (A) and *uptime* (U) values of all responding /24-blocks. We define host availability, $A(host)$ as the fraction of time a given host is on the network. We define host uptime, $U(host)$, as mean duration the address has a positive response, normalized by the duration of the censuses that are considered. This computation assumes that each probe is representative of the hosts stability for that entire census duration. (This assumption is not accurate for hosts that come and go frequently; we return to typical address occupancy durations Section 3.3.) We also define block availability and uptime, or $A(block)$ and $U(block)$, as the mean $A(host)$ and $U(host)$ for all responsive hosts in the block.

For a single census, $A(host)$ takes on a binary value for a positive reply or either negative or lack of reply, and $U(host)$ is not very meaningful. However, $A(block)$ is, by definition, an estimate of the fraction of hosts that are up in that block. If addresses are considered equivalent, it is also the probability that any addresses is occupied.

Figure 2 shows only addresses that have responded positively at some point. In fact, the majority of blocks are non-responsive, and so 8,256,560 blocks should appear at $(A = 0, U = 0)$.

These metrics are far from ideal: with observations every three months, our measures of availability and particularly stability are hugely under-sampled. In addition, the two measures are not completely orthogonal, since large values of $U$ can occur only for large values of $A$ and small values of $A$ correspond to small values of $U$. In fact, $U = A/N_U$ where $N_U$ is the number of uptime periods. Finally, taking the mean of all hosts in a /24 block may aggregate hosts with different purposes or administrators.

Despite this, this figure suggests some aspects of the Internet address space utilization. First, the vast majority of blocks are lightly utilized with low uptime, near ($A = 0$, $U = 0$). However, a few blocks are heavily utilized and always up (near $A = 1$, $U = 1$) Manual examination suggests that these blocks near ($A = 1, U = 1$) represent server farms, typically hosting many different web and mail sites. Second, blocks with a medium value of $A$ get pulled apart, where larger $U$ values suggest blocks with servers that turned on mid-way through our census, while smaller $U$ values suggest blocks blocks where hosts come and go frequently.

Finally, many hosts follow the $A = U$ diagonal. These hosts correspond to a single uptime occurrence, whether it's a server that is always up, or a non-responsive host that replied only once in all 14 censuses.

While these results make too many assumptions to be definitive, they demonstrate that a census can results about the Internet that seem consistent with our intuition. (We strengthen this claim in Section 8.1 by comparing census results to much more frequent survey.) We focus on two separate problems below: first, we sample this population, taking much more frequent probes of only about 1% the Internet to provide stronger statements about host stability (Section 3). Second, we look at long-term trends of hosts that fail to respond positively in censuses over three years to measure firewall deployment (Section 6).

# 3 Surveying a Fraction of the Internet Address Space

While a census captures the entire Internet, by necessity it visits any part of the Internet only occasionally. With four parallel probers we can take an Internet census in about three months. As shown in Section 2.4, this complete view provides a unique snapshot of the Internet. Yet our estimate of host uptime there is very questionable, since many hosts such as laptops and desktops that are turned off at night come and go on timescales of hours, so our census grossly undersamples this dynamic signal.

We therefore now turn to a *survey*-based methodology designed to probe individual addresses frequently enough to capture the behavior of dynamic hosts, and to be resilient to loss of individual probes.

Our methodology *samples* a fraction of the Internet and probes hosts in that sample repeatedly at a higher frequency and a shorter duration than a census. A primary challenge of a survey is to ensure that our sample is large enough to represent the Internet, that it is unbiased, and to understand what measurement uncertainty sampling introduces.

We have been conducting surveys of different samples of the address space since March 2006. Each survey begins concurrently with a new Internet census, but because surveys sample a smaller section of the address space at higher rates, they each last only one week. Table 2 lists the dates of our surveys so far.

|  |  | **Duration** | **Blocks** | **Blocks** |
|---|---|---|---|---|
| **Name** | **Start Date** | (days) | **probed** | **responded** |
| $IT^{\text{survey}}_{14w}$ | 2006-03-09 | 6 | 260 | 217 |
| $IT^{\text{survey}}_{15w}$ | 2006-11-08 | 7 | 23,482 | 17,528 |

Table 2: Summary of conducted surveys

We next describe our approach to surveying Internet addresses and present preliminary results.

## 3.1 Survey Design Issues

The key design issues in conducting a survey are setting the desired probe frequency at each address and selecting the sample of addresses to survey. We review these issue and implementation below.

**How many:** Our choice of how many hosts to survey was governed by several factors: we needed a sample large enough to be reasonably representative of the Internet population, yet small enough that we could probe each address frequently enough to capture individual host arrival and departure with reasonable precision. We studied probe frequencies as close as 5 minutes (details omitted due to space); based on those results we selected a probe frequency of 11 minutes as providing reasonable precision, and being relatively prime to common human activities that happen on multiples of 10, 30, and 60 minutes. We selected a survey size of about 1% of the allocated address space, or 24,000 /24 subnets in hopes of providing sufficient coverage of all kinds of subnets. We employ a single prober to survey this number of addresses. To pace replies, we only issue probes at a rate that matches the timeout rate, resulting in about 9,200 probes/second.

**Which addresses:** Given our target sample size, the next question is which addresses are probed. To allow analysis at both the host- and block-granularity we chose a clustered sample design where we fully enumerate each of 24,000 selected /24 blocks. In population surveys, clustered sampling is often used to reduced costs, whereas we use clustering to achieve different goals.

An important sampling design choice is the granularity of the sample. We probe blocks of addresses rather than individual addresses because numerically adjacent addresses often have similar properties. CIDR [12] and BGP routing exploit common prefixes to reduce routing table sizes, and so numerically adjacent addresses are often assigned to the same administrative entity. Because numerically adjacent addresses are often routed similarly we expect that they often share similar patterns of packet loss. To the extent blocks are managed similarly, probing an entire block makes it likely that we probe both network infrastructure such as routers or firewalls, and edge computers. We select a survey block size of hosts with the same 24-bit prefix, since that corresponds to the minimal size network that is allowed in global routing tables and is a common unit of address delegation.

We had several conflicting goals in determining which blocks to survey. An unbiased sample is easiest to analyze, but blocks that have some hosts present are more interesting, and we wanted to ensure we sample unusual parts of the Internet address space. We also wanted some blocks to remain stable from survey to survey so we can observe their evolution over time, yet it is likely that some blocks will “fail” over time, either becoming firewalled, being removed, or going dark due to renumbering.

Our sampling methodology design attempts to balance these goals. We use three different policies to select which address blocks will be probed: stable/random, stable/spaced, and novel/random. Half of the blocks are selected with a stable policy, which means that we selected them when we began surveys in September 2006 and retain them in future surveys. We selected the stable set of blocks based on $IT_{13w}$. Of the stable set of blocks, half of those (one quarter of blocks in the entire survey) were selected randomly from all subnets that had any positive responses. This set is relatively unbiased (affected only by our requirement that the block show some positive response). The other half of stable blocks were selected to uniformly cover all values of the Figure 2’s A-U plot[1] This set is strongly biased but ensures that we have representatives from even unusual blocks, including fully-populated, always up server farms and frequently changing dynamically addressed areas.

The other half of blocks are selected randomly, each survey, from the set of /24 blocks that responded in the last census. We chose this selection method to provide unbiased coverage of all the address space while making it likely that we will get responsive blocks. This preference for responsive blocks does however bias our selection to favor the actively used part of the address space, and it insures that we will not see any “births” of newly used blocks in our survey data.

In spite of these techniques, we actually see a moderately large number (27%) of unresponive blocks in our surveys, suggesting address usage is constantly evolving.

**How long:** We intend to collect survey data for two week periods. We chose one week as sufficient to capture two workday/weekend cycle, while hopefully not so long as to be burdensome to the observed subnets. Our surveys to date (Table 2), however, have only captured periods of about one week.

## 3.2 Survey Implementation

Our basic survey software implementation is almost identical to that used for conducting a census (Sections 2.1 and 2.2). The only difference is that we filter outgoing probes by some pre-selected sample of the address space. As an optimization, rather than do this filtering each pass, we compute it once and record the randomized probe order. Thus the prober can simply replay the probes as fast as feasible, limited by a fixed number of outstanding probes (to cap internal state) and an selected maximum probe rate (to cap bandwidth consumption).

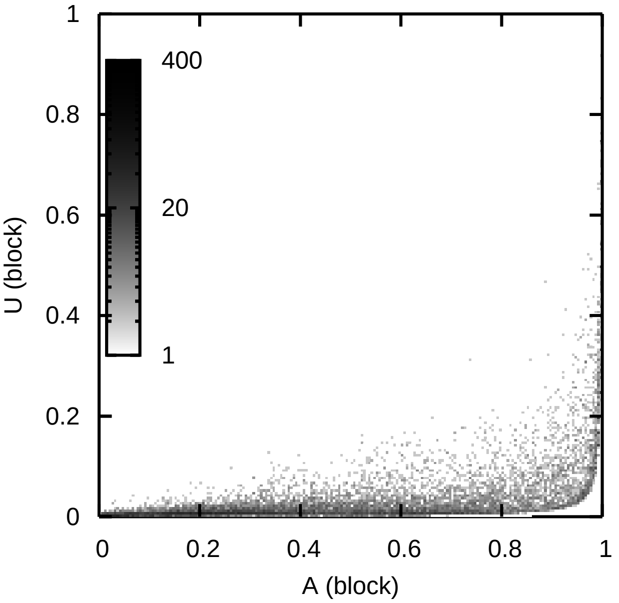


Figure 3: Density of /24 Internet address blocks in survey $IT^{\text{survey}}_{15w}$, grouped by block availability and block uptime.

Since filtering walks every address approximately every 11 minutes, a given block could see bursts of up to 254 probes. To reduce this effect, we pace probes across the 11 minute window, so any particular /24 block should see a probe once every 2–3 seconds on average.

## 3.3 Preliminary Results

By including frequent probing of a few addresses, survey data complements our censuses by capturing address occupancy at much finer timescales.

**Re-characterizing Internet Address Blocks:** We first revisit our census-derived $(A,U)$ plot (Figure 2) to see how more precise uptime measurements change our view.

Figure 3 shows a block-level $(A,U)$-plot from $IT^{\text{survey}}_{15w}$. Several shifts are apparent from our census-based plot. First, this data confirms that the general probability mass is near $(A = 0, U = 0)$. However, rather than a strong $A = U$ line, most of the weight is on the $U \simeq 0$ line, suggesting in sparsely populated subnets most hosts are unavailable. This shift is partially a result of block-aggregation; a stronger diagonal returns when we look at $(A(host), U(host))$ in Figure 8. Second, we observe that heavily populated blocks show much higher uptime values. Partly this relationship follows because $A$ and $U$ are not orthogonal, but it also suggests that the Internet is more dynamic than one might have expected. Finally, a single brief outage halves the $U$ value; although we can partially correct for this through loss repair (Section 7.3).

**Typical duration of address occupancy:** Finally, our survey data allows us to suggest to what extent addresses on the Internet are used dynamically. There are many reasons to expect that most hosts on the Internet are dynamically addressed, since many end-user computers use dynamic ad-

[1] Actually, this selection used a earlier version of the A-U plot, an A-V plot, where V was *volatility*, defined as the number of times the address transitioned between responsive to non-responsive. Volatility and uptime are related functions (extreme uptime implies low volatility), so we believe this slightly different formulation still covers the A-U space.

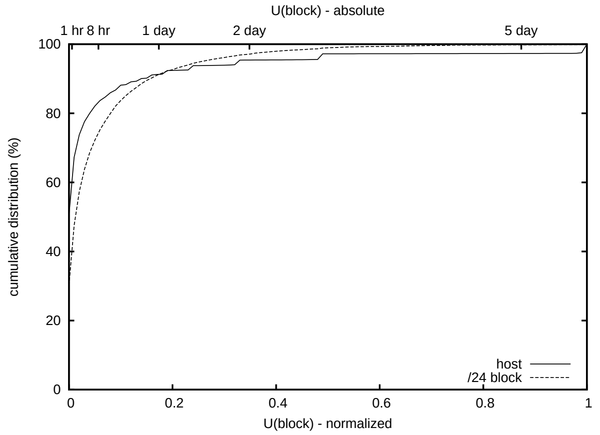


Figure 4: CDF of $U(host)$ and $U(block)$ from 1-repaired Survey $IT^{\text{survey}}_{15w}$.

dress assignment, either because they are mobile and adopt the address of their current location, or because ISPs change client addresses to discourage home servers, or to provide static addressing as a value- (and fee-) added service. In addition, some hosts may appear dynamic because they are regularly turned off.

Figure 4 shows distribution of host uptimes (with 1-repair Section 7.3). This data shows that vast majority of hosts are not particularly stable, and are up for a fraction of the observation time. We see that 50% of hosts are up for 81 minutes or less. A small fraction of hosts, however, are quite stable, with about 3% up almost all the time, and another 8% showing only a few (1 to 3) brief outages. We use this relationship to bound the number of servers in the visible Internet in Section 5.

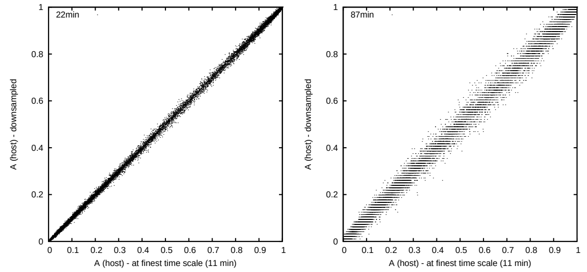
## 4 Census and Survey: trading time and space

With more more than four billion potential IPv4 addresses, full enumeration is not easy, but also not impossible. However, extremely frequent probing is resource-intensive at the origin site, and may be misinterpreted as offensive at the destinations, so one may choose to probe at a slower rate than is technically possible. Moreover, for any fixed probe rate, there is an an inherent tradeoff between how often a given address examined and how many addresses are considered.

This section explores this fundamental tradeoff: we can probe a few addresses at a high rate, but, as we increase the number of destinations, the maximum probe rate decreases linearly. In this section we're trying to explore the relative advantages of higher rate and larger probed address space and the effect of this tradeoff to the accurateness of measurements.

### 4.1 Sampling in Time

As Internet addresses can be probed at different rates, we would like to know how the probe rate affects the fidelity of our measurements. Increasing the sampling rate, while keeping the observation time constant, should give us more samples and hence a more detailed picture. However, probes that are much more frequent than changes to the underlying phenomena being measured cannot further improve accuracy (due to the Shannon sampling theorem [33]). Unfortunately, we do not necessarily know the timescale of what we observe. In this section we therefore evaluate the effect of changing the measurement timescale on our $A(host)$ metric.



(a) 2x downsampling

(b) 8x downsampling

Figure 5: Effect of downsampling fine timescale $A(host)$. Data from $IT^{\text{survey}}_{15w}$.

In order to examine what effect the sampling interval has on the fidelity of our metrics we simulate the effect of varying the probe rate by decimating this fine timescale dataset. We treat the complete dataset as ground truth, then we throw away every other sample to halve the effective sampling rate. Applying this process repeatedly gives exponentially coarser sampling intervals.

Figure 5 shows the results of two levels of downsampling for every host that responds in our fine timescale survey. In the figure, each host is shown as a dot with coordinates representing its accessibility at the finest time scale ($x$-axis) and also at a coarser timescale (the $y$-axis). If a coarser sample provided exactly the same information as finer samples we would see a straight line, while a larger spread indicates error caused by coarser sampling. In addition, as sampling rates decrease, data collects into bands, because $n$ probes can only distinguish $A$-values with precision $1/n$. As we increase the sample interval from 2- to 8- and 16-times the finest sampling, we observe that the width of the bar grows.

While these graphs provide evidence that sparser sampling increases the level of error, they do not directly to quantify that relationship. Instead, we group hosts into bins based on their $A(host)$ value at the finest timescale, then compute the standard deviation of $A(host)$ values in each bin as we reduce the number of samples per host. This approach quantifies the divergence from our ground-truth finest timescale values as we sample at coarser resolutions. Figure 6 shows these standard deviations for a range of sample timescales, plotted by points. This graph clearly shows that coarser sampling corresponds to wider variation in the measurement compared to the true value. We see that the standard deviation is the greatest for hosts with middle values of $A$ (local maximum around $A = 0.6$) and significantly less at the extreme values of $A = 0$ and $A = 1$.

To place these values into context, assume for a moment that host occupancy is strictly probabilistic, and that a host is present with probability $p$. Thus $E(A(host)) = p$, and each measurement can be considered a random variable $X$ taking values one or zero when the host responds

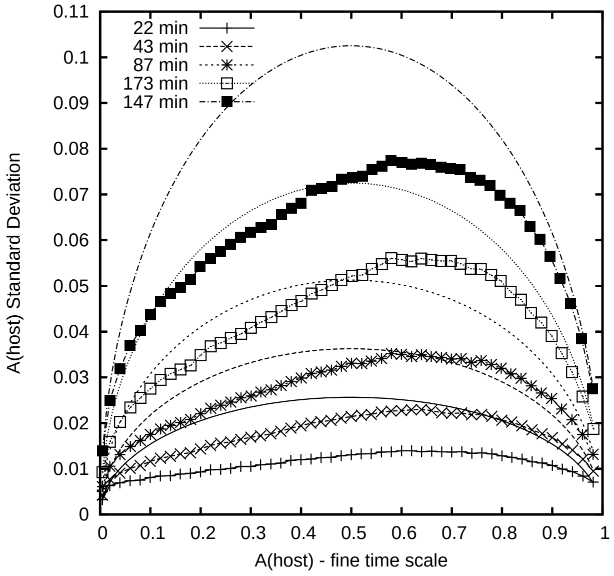


Figure 6: Standard deviation (from $IT^{\text{survey}}_{15w}$) as a function of ground truth $A(host)$ metric (from $IT_{15w}$), with theoretical 90% confidence intervals.

(with probability $p$) or is non-responsive (with probability $1-p$). With $n$ samples, we expect $np$ positive results, and $\hat{A}(host)$ will follow a binomial distribution with standard deviation $\sqrt{np(1-p)}$. On these assumptions, we can place error bounds on the measurement: our estimates should be with in $\hat{A}(host) \pm 1.645\sqrt{\hat{p}(1-\hat{p})/n}$ for a 90% confidence interval. The curves $\sqrt{\hat{p}(1-\hat{p})/n}$ are also shown in Figure 6 as lines. We can see that the measured variance is nearly always below the theoretical prediction. This reduction is potentially caused by correlation between hosts in same block. The prediction becomes more and more accurate as we increase the time scale and samples become more "random" approaching the binomial distribution.

These results assume our measurements are unbiased. This assumption is not strictly true; we explore the bias induced by probe loss in Section 7.3.

## 4.2 Sampling in Space

We can survey an increasing number of hosts, but only at a diminishing rate. In the extreme case of our census, we probe every host only once every several months. Data so sparse makes interpretation of uptime highly suspect, because measurements are taken much less frequently than the known arrival and departure rates of hosts such as mobile computers. Much more frequent sampling is possible when a smaller fraction of the Internet is considered, however this step introduces sampling error. In this section we review the statistics of population surveys to understand how this affects our results. The formulae below are taken from Hedayat and Sinha [17]; we refer interested readers there.

When we consider the need to find the proportion of the population that meets some criteria, such as the mean $A(host)$ values for the Internet, we draw on two prior results of simple random sampling. First, a sample of size $n$ approximates the true $A$ with variance $V(\hat{A}) \simeq A(1-A)/n$ (when the total population is large, as it is in the case of the IPv4 address space). Second, we can estimate the margin of error $d$ with confidence $1-\alpha/2$ for a given measurement as:

$$d = z_{\alpha/2}\sqrt{A(1-A)/n} \quad (1)$$

when the population is large, where $z_{\alpha/2}$ selects confidence level (1.65 for 95% confidence).

Second, when estimating a non-binary parameter of the population, such as mean $A(block)$ value for the Internet with a sample of size $n$, the variance of the estimated mean is $V(\bar{A}(block)) = S^2_{\bar{A}(block)}/n$, where $S^2_{\bar{A}(block)}$ is the true population variance.

We draw our conclusion from the work in human sampling work: by controlling the sample size we can control the variance and margin of error of our estimate. Our selection of 1% of the Internet address space provides good bounds for our purposes.

# 5 Bounding the Number of Servers on the Internet

One application of fine timescale probing of Internet hosts is to attempt to estimate the number of servers on the Internet. We characterize servers on the Internet as any host that is highly and consistently available; that is, an address that has a high host availability and host uptime values as defined in Section 2.4. We know that servers are not the *only* machines that are always on, since many hosts used primarily as clients are also accessible on the visible Internet at all times, and routers are always accessible but are not generally considered servers. We therefore consider highly accessible hosts to represent an upper bound on the number of possible Internet servers.

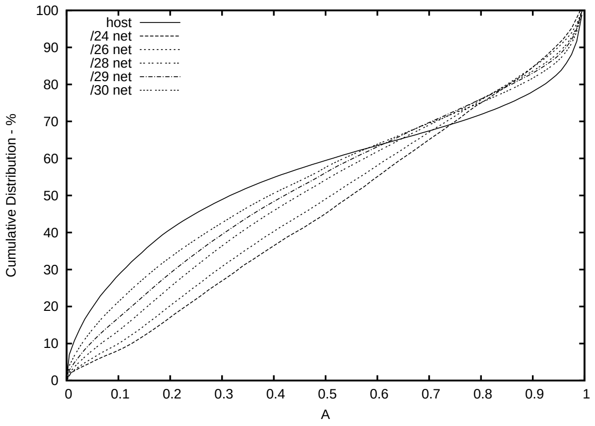


Figure 7: CDF of $A(host)$ and $A(block)$ from from $IT^{\text{survey}}_{15w}$.

Figure 7 shows the cumulative density function of A values for hosts, and for different size blocks, computed over all survey $IT^{\text{survey}}_{15w}$. The hosts clustered around the two peaks on the figure represent clear cut case stable and unstable hosts. We can conclude that hosts accessible less than 10% of the time are clearly not servers. We define hosts with 95% availability or better to be *very stable hosts*. Based on our assumption that stable hosts provide an upper bound on the number of servers, this data suggests that 16.4% of responsive hosts in the survey are servers.

We can next project this estimate to the whole Internet by extrapolating from the survey to the census taken over the whole Internet. Our survey finds 1.75M responsive hosts in 17.5k responsive /24 blocks, suggesting a mean of 16.4 servers per responsive block. The corresponding census finds 2.1M responsive blocks, suggesting an upper bound of 34.9M servers in the entire Internet. This estimated upper bound depends on mapping between survey and census; we return to that relationship in Section 8.1.

We considered and dismissed what seems like a simpler extrapolation. Given 103M responsive hosts in our census, we could estimate that 16.4% of these, or 16.8M addresses, are potential servers, However, this estimate does not account for the fact that our survey was biased (by only choosing to survey blocks that were responsive in the prior census, not all allocated blocks), and our survey is much more robust to packet loss, since each address is probed more than 916 in a one weeks survey rather than once in the three month census.

Finally, we know that this estimate is a loose upper bound on servers we measure very available hosts, such hosts also include many clients and routers. In Section 7.2 we quantify the number of visible multi-homed hosts, many of which are likely routers, and show that about 6% of hosts in our census are multi-homed.

It is remarkable that hosts in-between are a significant percentage of the whole. In fact, 55% of addresses fall between 10% and 95% availability. It would be easy to dismiss these hosts as moderately reliable clients, however an alternate explanation is that they are servers that are turned on in the middle of our survey. To understand if this alternative explanation is possible, we turn to the host-uptime metric from Section 2.4, which quantifies the duration a host is available.

The value of $U(host)$ allows us to distinguish between addresses that are intermittently occupied 50% of the time, and those that are unoccupied for one week, then consistently occupied for the following week. Both such hosts will show $A(host) = 0.5$, in the first case the $U(host)$ value will approach zero, while the second will show $U(host) = 0.5$ . (In this case, the minimum value $U(host)$ would occur when probes alternate ACK and NACK; with 11 minute samples over one week that would indicate $U(host) = 0.001$.) We therefore conclude that addresses where $U(host) = \phi A(host)$ suggest a late-arriving server when $\phi$ is large, say 0.9, while other cases suggest a flaky client.

Figure 8 shows how hosts are distributed on the as defined by their $(A(host), U(host))$ values. (By comparison, Figure 3 shows $(A(block), U(block))$, decreasing the prominence of the $A = U$ line.) We can see that all intermediate values of $A$ have very low corresponding $U$-values, which can only mean that intermediate $A$-value servers are very infrequent. In fact, only 1% of addresses have $0.4 < A(host) < 0.8$ and $U(host) > 0.3$. From this, we dismiss our alternative explanation of late-arriving servers and state more definitively that addresses with $A(host) > 0.95$ are likely servers.

Finally, based on this survey, we can project to the total number of servers in the Internet with margin of error. We surveyed 1% of the total responsive address space and observed 286,550 stable hosts. This value and the reasoning above implies at most 35.7M potential servers in the visible Internet. The margin of error from sampling is about 0.4%. Additionally, our analysis above suggested that up to 1% of addresses have $A(host) < .95$ and moderate or large $D(host)$ values, suggesting potential server births during observation. This implies our bound may be 1% low.

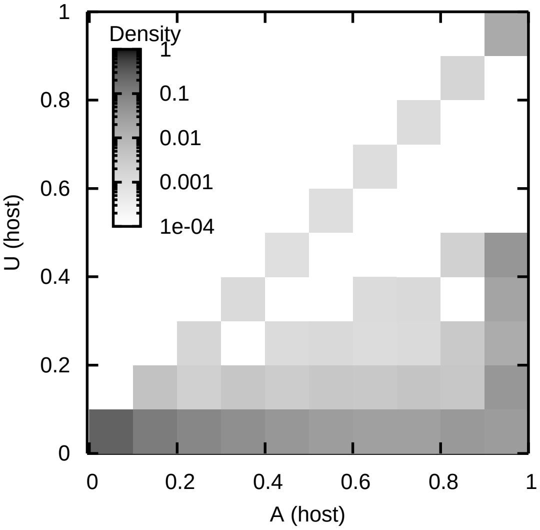


Figure 8: Density of hosts from $IT^{\text{survey}}_{15w}$.

Two caveats about this upper bound: First, it is a loose upper bound, because it measures hosts that are 95% stable or more, but not all stable hosts are servers. Second, however we omit servers that do not respond to ICMP echo requests. We know that such servers exist, since particularly security-conscious server operators may allow access via to the service itself, but not respond to pings. However, we believe this case applies to relatively few servers. Quantifying these servers is an area of future work.

## 6 Trends in Firewall Deployment

Large numbers of Internet hosts lie behind firewalls, which are configured to restrict, block or rate-limit traffic according to private local policies. Firewalls clearly affect the visibility of hosts to censuses. In this section we study trends in the deployment of visible firewalls over 15 months to begin to understand their effect on our observations.

Counting the number of hosts behind firewalls is difficult since the goal of a firewall is often to shield hosts from external access. Measuring firewalls themselves is also difficult because many firewalls simply drop packets, making them invisible to our probing. Some firewalls, however, respond to ICMP echo requests with negative acknowledgments that include codes designating that communication is "administratively prohibited". We use this information to estimate the number of firewalls and firewalled hosts.

We begin with some terminology and definitions. We define a firewall as a software or hardware device that intentionally hides, from our probes, an active network interface that is otherwise connected to the public Internet and assigned a public IP address. (Since our focus is the public Internet, we do not attempt to measure hosts behind NATs with private IP addresses.) A firewall is a device or software that can intercept packets before they reach a set of target hosts. With regard to our probes, *silent firewalls* discard the probe without reply, while *visible firewalls* generate a reply that indicates communication is administratively prohibited. Many host operating systems include firewall capabilities that protect a single machine. We call these *personal firewalls*, in contrast to *subnet firewalls* which are typically implemented by routers, PCs or dedicated appliances and cover a block of addresses. When appropriate, we use the term firewall to cover all these different devices and software.

In this section, we use our datasets to count the visible firewalls in the Internet, both personal and subnet firewalls, and estimate the address space they cover. We estimate the total firewalled address space, the total number of firewalled address blocks, and the total number of personal firewalls in the Internet. Because we miss silent firewalls, these measurements provide only lower bounds. Finally, we analyze trends in firewall deployment over a 15-month period covered by censuses $IT_7$ through $IT_{15w}$ (all censuses that recorded NACKs).

## 6.1 Methodology

Our methodology to count firewalls is as follows. For each Internet census we probe all addresses with ICMP echo requests. We consider both the source and the content of the reply to evaluate presence of a firewall. If the response is destination unreachable (type 3) indicating network, host, or communications administrative prohibited (codes 9, 10, and 13, respectively), we infer that the response comes from a visible firewall.

We then compare the probed address $P$ to the source address of the reply message $R$. When $P = R$, the host itself replied, and so we classify $P$ as a personal firewall. When $P \neq R$, then we conclude that a subnet firewall with address $R$ replied on $P$'s behalf. We can then examine all probed addresses $P_i$ with the same reply address $R$ to determine the coverage of subnet firewall $R$, its *firewalled block*. We analyze our censuses to estimate the number of firewalled addresses, the number of firewalled blocks, their distribution by size and their evolution over time.

We estimate the size of a firewalled block by grouping addresses according to the replying addresses $R$. A block firewalled by $R$ is the largest $[l,h]$ address range such that $\forall$ $p \in [l,h]$, a probe to address $p$ elicits a an administratively prohibited reply from $R$, or a positive reply (echo reply, type 0) from $p$, or there is no response for probe to $p$. This definition of firewalled blocks tolerates lost probes (by ignoring non-responses) and considers the common practice of allowing a few publicly-visible hosts (often web servers) in the middle of an otherwise firewalled range of addresses.

## 6.2 Results

We begin by considering the size of the firewalled address space. Figure 9 shows the absolute number of addresses protected by visible firewalls, and the ratio of that count to the number of responsive addresses. We calculate the number of firewalled addresses by summing up the addresses contained in all firewalled blocks.

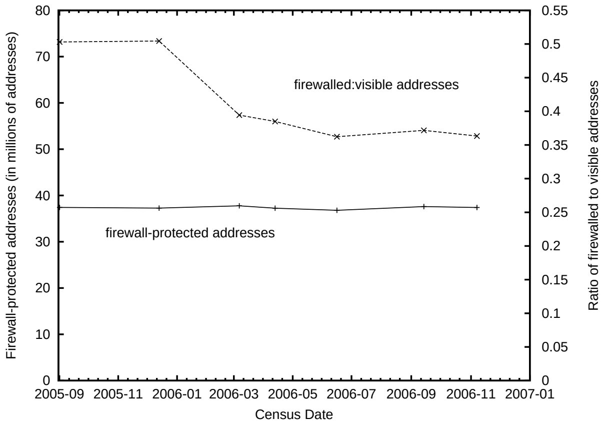


Figure 9: Number of addresses protected by visible firewalls (including personal firewalls), in absolute terms (left scale) and in ratio to visible, non-firewalled addresses. (Data from $IT_7$ through $IT_{15w}$.)

We see that there are nearly 40M addresses protected by visible firewalls. The firewalled address space is a very small fraction of the allocated address space (2.6B–2.8B addresses). The firewalled address space is, surprisingly, relatively stable over this period. However, when we compare the ratio of addresses protected by visible firewalls to the number of responsive, non-firewalled addresses, we see a moderate downward trend. In mid-2005, there was 1 visibly firewalled address for every 2 responsive addresses; by the end of 2006 this ratio had declined to nearly 1:3. We suspect that this trend is due to an increase in the number of invisible firewalls (firewalls that simply discard probes); further investigation is required.

We next turn to firewall block size, the address space covered by each firewall. First, we observe that there are very many personal firewalls. We see between 190,000 and 224,000 across our surveys, with no consistent trend over time. Personal firewalls greatly outnumber subnet firewalls, 4:1.

Turning to subnet firewalls, Figure 10 shows the cumulative distribution of sizes of firewall blocks, omitting personal firewalls. We assume that the number of blocks corresponds to the number subnet firewalls. We see bumps at block sizes that are powers of two, with a pronounced bump at /24 subnets, but interestingly, at /29 and /30 subnets. We also notice a slight increase in the number of blocks over the course of our study, but mostly due to additional firewalls covering one address each. Finally, while personal firewalls outnumber subnet firewalls, the latter cover the vast majority (more than 99%) of the firewalled address space.

From these observations we make several conjectures about trends in firewall use. Since we see little increase in the number of firewalled hosts across our censuses, we conjecture that most newly deployed hosts are either visible, or

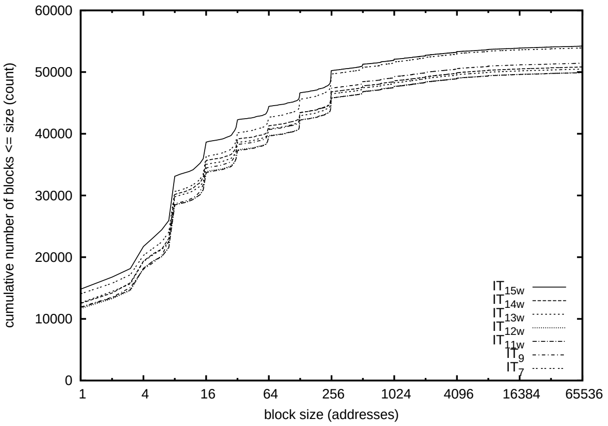


Figure 10: Cumulative distribution of firewalled blocksize. This graph omits 190–225k personal firewalls. (Data from $IT_7$ through $IT_{15w}$.)

go behind silent firewalls that our methodology is unable to account for. Given the relative stability in the number of visible firewalls, we conjecture that existing firewalls maintain visibility and most new firewalls are configured to be invisible. The latter may reflect the heightened sense of security in new deployments, while the former the inertia in changing existing configurations.

## 7 Limitations of Active Probing

While an Internet census captures information about the entire Internet, and our surveys capture more detailed information about a fraction, both have clear limitations in their applicability. We review those here: invisible hosts, multiply-visible hosts, and probe loss.

### 7.1 Invisible Hosts

The most significant limitation of our approach is that we can only see the *visible* Internet. Hosts that are hidden behind ICMP-dropping firewalls and in private address space are completely missed; those behind NAT boxes appear to be at most a single occupied address.

Approaches to characterize the extent of the invisible Internet are an area of future work; in this paper we claim to only identify visible addresses. In Section 6 we look at visible firewall deployment, but we cannot quantify the size of the invisible network.

### 7.2 Multi-homed hosts

We generally assume that each host occupies only a single IP address, and so each responsive address implies a responsive host. This assumption is violated in two cases: some hosts have multiple public network interfaces, and some hosts use different addresses at different times.

Multiple public IP addresses for a single host are known as *aliases* in Internet mapping literature [14], and several techniques have been developed for *alias resolution* [14, 38] to determine when two IP addresses belong to the same host.

One such technique is based on the fact that some multi-homed hosts or routers can receive a probe-packet on one interface and reply using a source address of the other [14]. The source address either is fixed or determined by routing. This behavior is known to be implementation-specific.

This technique is particularly suitable for large-scale Internet probing because it can be applied retroactively. Rather than sending additional probes, we re-examine our existing traces to find responses sent from addresses different than were probed. We carried out this analysis in one of census $IT_{15w}$ and found that 6.7 million addresses so responded, a surprisingly large 6.5% of the 103M total responses.

In addition to hosts with multiple concurrent IP addresses, many hosts with have multiple sequential IP addresses, either due to mobility, or due to changing DHCP assignments. In general, we cannot track this since we only know *address* occupancy and not the identity of the occupant. However, Section 5 showed how host-uptime $U(host)$ varies, and our data can project typical host lifetimes. Further work is needed to understand the impact of hosts that take on multiple IP addresses over time.

### 7.3 Probe Loss

An important limitation of our methodology is our inability to distinguish between host unavailability and probe loss. Probes may be lost in several places: in the LAN or an early router near the probing machine, in the general Internet, or in the edge near the destination. In this section, we examine how lost lost probes affect observed availability and the distribution of $A(host)$ and $A(block)$.

We minimize chances of probe loss near the probing machines through two different ways. First, we rate-limit outgoing probes to so that it is unlikely that we overrun nearby routers buffers. Also, our probers checkpoint their state periodically and so we are able to stop and resume probing for known local outages. Finally, when local outages were detected only after-the-fact, we can examine the traces for numbers of consecutive non-response and restart the trace from state before the outage.

We expect three kinds of potential loss in the network and at the far edge: occasional loss due to congestion, burst losses due to routing changes [20] or edge network outages, and burst losses due to ICMP rate-limiting a destination last-hop router. The key approach to managing these kinds of loss is our pseudo-random probing order (Section 2.2). With the highest probe rate to any /24 block of one probe every 2–3 seconds in a survey, or 9 hours for a census, rate limiting should not come in to play. In addition, with a census, probes are spaced much further apart than any kind of short-term congestion or routing instability, so we rule out burst losses for censuses and concerned only with random loss.

Random loss is of concern, however, because the effect of loss is to *skew* the data towards a lower availability. This skew differs from surveys of humans where non-response is apparent and alter data interpretation, and where survey error is often equally in the positive and negative directions. We expect to see random loss rates of a few percent due to network congestion (for example, studies with TCP have suggested 90% of connections have at most 5% loss [1]).

We account for loss differently for censuses and surveys. For surveys we *repair* random loss as described below. For censuses, data collection is so sparse that repair is not appropriate. There, we focus on $A(block)$ rather than $A(host)$. By

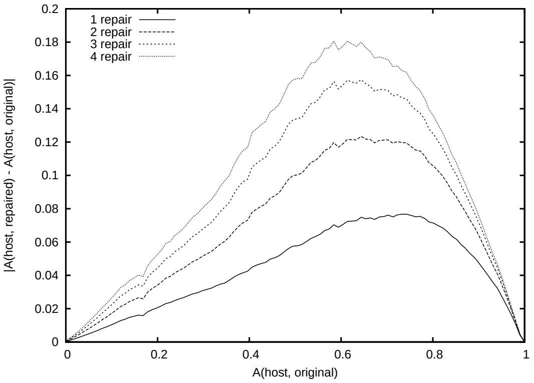


Figure 11: Distribution of differences between the *k*-repair estimate and non-repaired $IT^{\text{survey}}_{15w}$.

averaging responses for an entire block of addresses, random loss of any individual probe has less impact.

For surveys, we attempt to detect and repair random probe loss through a *k*-repair process. We assume that a random outage causes up to *n* consecutive probes to be lost. We repair losses of up to *k* consecutive probes by searching for two positive responses separated by up to *k* non-responses, and replacing this gap with assumed positive responses. We can then compare $A(host)$ values with and without *k*-repair; clearly $A(host)$ with *k*-repair will be higher than without.

Figure 11 shows how much *k*-repair changes measured $A(host)$ values for $IT^{\text{survey}}_{15w}$. Larger values of *k* result in greater changes to $A(host)$; but the change is fairly small: it changes by at most 10% with 1-repair. We also observe that the change is largest for intermediate $A(host)$ values (0.4 to 0.8). This skew is because of the definition of $A$: highly available hosts ($A(host) > 0.8$) have very few outages to repair, while mostly unavailable hosts ($A(host) < 0.4$) have long-lasting outages that cannot be repaired.

Finally, although we focused on how loss affects $A(host)$ and $A(block)$, it actually has a stronger effect on $U(host)$. Recall that $U$ captures host continuous uptime. For a host up continuously $d_0$ days has a $U(host) = 1$, but an brief outage anywhere after $d_1$ days of monitoring gives a mean uptime of $(d_1 + (d_0 - d_1)/2$ days and a normalized $U(host) = 0.5$, and a second outage reduces $U(host) = 0.33$. This level of outage contributes to the differences between Figure 2 and 3, where fine-timescale measurement gives many more opportunities for outages. This effect can be partially mitigated with *k*-repair, but we are also considering alternative metrics of uptime.

# 8 Result Validation

The results in prior sections have assumed our measurements are not systematically biased. In this section we consider several potential biases, including very long and sparse coarse-timescale measurements and location of the probing machines, concluding that neither significantly changes our results.

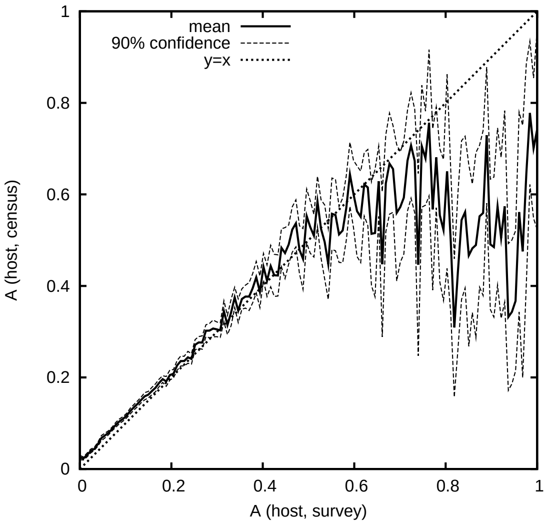


Figure 12: Comparison of $A(block)$ for coarse and fine time scale data for $IT_{15w}$ and 1-repaired $IT^{\text{survey}}_{15w}$.

## 8.1 Comparing Coarse and Fine Timescale Measurements

A significant difference between our census and surveys is the timescale of measurement: a census probes a given address every 3 months, while a survey every 11 minutes. Thus while it makes sense to treat a survey's consecutive probes of the same address as a timeseries, it is more difficult to evaluate evolution across censuses because long-term host changes (renumbering and host birth and death) are significant. In addition, loss repair is not generally possible

However, we can compare a concurrent census and survey to gain some validation of their accuracy. Because $A(host)$ is poorly defined for a single census, we compare $A(block)$ for /24 blocks in $IT_{11w}$ and $IT^{\text{survey}}_{11w}$.

To compare census and survey, we arrange all blocks by increasing $A(block)^{survey}$ computed from 1-repaired survey data. Since this survey represents 916 probes of each address spread over one weeks, we consider this ground truth. We then group subnets that have similar $A(block)^{survey}$ values, gathering 254 integral "bins" with about that number of responsive hosts in the block. Finally we calculate the corresponding $A(block)^{census}$ from census data for the same subnet. In each $A(block)^{survey}$ bin we therefore get some number of similar $A(block)^{census}$. From these values we plot the mean and 90% confidence intervals of $A(block)^{census}$.

This comparison is shown for $IT_{15w}$ and $IT^{\text{survey}}_{15w}$ in Figure 12. Ideally the means should match the diagonal and confidence intervals should be zero. We see a reasonable match (the correlation coefficient is 0.74). The values are close for blocks with lower availability ($A < 0.5$), but we see that the census under-estimates $A(block)$ value for higher availability blocks.

We believe the match is poorer for large $A$ values because there are many stable blocks with only one or two stable hosts. If a census misses one of these hosts due to probe loss, that block will show very high error. Two other poten-

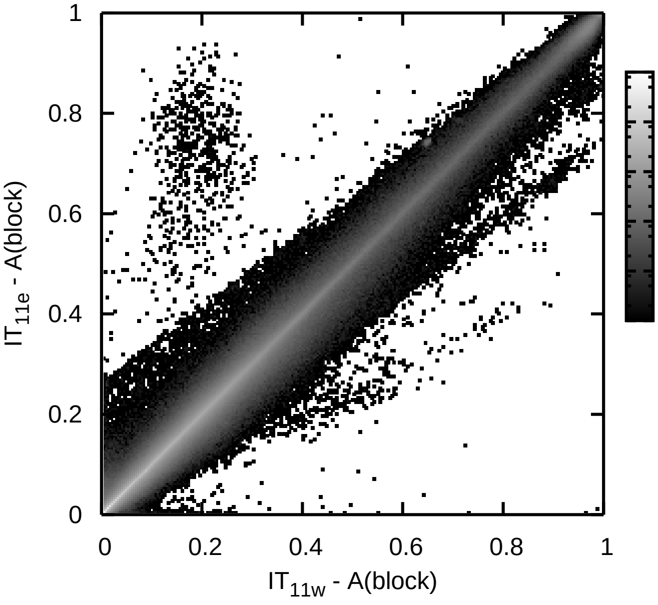


Figure 13: Subnets' A values from two censuses taken from widely different locations: $IT_{11w}$ and $IT_{11e}$.

tial causes are that survey $IT^{\text{survey}}_{15w}$ lasted only 6 days, from Wednesday through Tuesday. It may be that more hosts are more frequently unavailable on weekends. A final possibility is that our survey's probe rate of 1 probe every 2–3 seconds is too high and is triggering block-level ICMP rate-limiting.

Because census estimates of $A(block)$ are relatively sparse, we had some concern that they might be overly altered by loss. From this comparison we conclude that block-level estimates are quite similar from both a census and a survey, providing confidence in the accuracy of $A(block)^{census}$

## 8.2 Measurement Location

A second possible source of bias is measurement location. Our probers are all in the same place in the Internet; it may be that this location provides a poor view of parts of the Internet, perhaps due to consistently congested links or incomplete routing.

To rule out this source of potential bias, censuses since March 2006 have been done in pairs from two different locations. A "West" census $IT_{11w}$ is taken from the ISI network in Marina del Rey, California; while an identical census $IT_{11e}$, is taken from the ISI's East-coast office in Arlington, Virgina. We use a different seed value at the different sites so probe order is different, although concurrent. These sites have completely different network connectivity.

Figure 13 compares the $A(block)$ values measured from each vantage point in a density plot. We see the vast majority of /24 blocks line on the $x = y$ axis as we would expect, with a few outliers.

In order to quantify the differences between these two censuses, we plot a PDF (relative frequencies) of the difference of $A(block)$ measured from each site. We omit this graph due to space, but it shows a strongly Gaussian distribution, with nearly all $A(block)$ differing by less than 5%.

From this comparison we conclude that our results are not significantly altered by a change in location.

## 9 Related work

To our knowledge, there has been no recent work to characterize the behavior of Internet edge hosts by active probing. there has been no recent work that has attempted to characterize the availability of hosts in the Internet or of the prevalence of firewalls, using either a census or a survey.

To our knowledge, there has been no recent work that has attempted to characterize the availability of hosts in the Internet or of the prevalence of firewalls, using either a census or a survey. Of course, Internet address space surveys have been conducted periodically [6] but these surveys have focused on enumerating the number of Internet hosts by traversing the DNS. This approach does not give insights into host availability or firewall prevalence. Earlier surveys in this series estimated host counts by pinging randomly chosen IP addresses. However, the Internet has grown significantly since these surveys and it is unlikely that their results hold today.

Closest to our methodology of active probing are the several projects that measure Internet connectivity, including Rocketfuel [38], Mercator [14], Skitter [18], and Dimes [34]. The primary goal of these projects is to estimate the macroscopic, router-level connectivity of the Internet. These project therefore do not exhaustively probe edge-hosts in IP address space, instead using tools such as traceroute to edge addresses to collect data about routers that make up the middle of the Internet.

Several other efforts have shared our goals of studying properties of the IP address space, but use different methodologies than we do.

Meng *et al.* use BGP routing tables have been used to study IPv4 address allocation at the block level [23]. Like our work, this work is a longitudinal study of address space allocation, in this case, for seven years. However, their approach considers only block-level information gathered from IANA and injected into the global routing tables, not a host-level study, and they consider only new blocks, not the entire IPv4 address space.

As another example, Kohler *et al.* [19] studied the structure of IP destination addresses seen through passive observations on Internet links. Their measurements were conducted at a few locations that included access links to universities, ISP routers with local peerings, and a major ISP's backbone routers. Their data collection considered several links, each measured for several hours, observing between 70,000 and 160,000 addresses. They discover multifractal properties of the address structure and propose a model that captured many properties in the observed traffic. By contrast, our census unearthed upwards of 50 million distinct IP addresses through active probing of addresses and so focuses more on the static properties of address usage rather than their dynamic, traffic dependent properties.

Finally, Narayan *et al.* propose a model of IP routing tables based on allocation and routing practices [25] , and Huston citeHuston and Gao *et al.* [5] (among others) have measured the time evolution of BGP tables and address space. This work focuses on BGP and routing, not the the temporal aspects of address space usage that we consider.

Because compromised home private machines are the source of a significant amount of unsolicited e-mail, a number of anti-spam organizations maintain lists of dynamically assigned addresses (examples include [37, 26]). This work complements our approaches to infer dynamic address behavior through fine-timescale probing, but uses primarily static or manually entered data, or semi-automated probing in response to spam.

Very recently research has explored how to detect dynamic address space usage by examining login rates to a major on-line e-mail hosting service [43]. As with our work they characterize IP address usage, however their methodology is based on passive monitoring of a large web service. Their work complements ours in that they can reach strong conclusions about the addresses that contact their service, but they cannot evaluate addresses that are omitted.

Much of the previous work on firewall detection has focused on discovering stealth firewalls. One of the earliest works was published on the Phrack website [10] and detected firewalls that did not verify checksums. Tools such as p0f [27] and nmap [31] have options to detect a firewall either by passively monitoring flows or actively sending specially crafted packets and analyzing responses. This work is orthogonal to ours since in this study we are only interested in visible firewalls.

## 10 Future Work

There are several directions for future work.

We began to explore how uptime can characterize servers and clients based on how stable their address use is. Significantly more work is needed to validate this approach and to more thoroughly characterize how dynamic addresses are typically used.

To minimize time to walk the entire Internet we chose not to attempt to retry dropped addresses (Section 2.1), and instead we correct for loss after-the-fact (Section 7.3). It would be interesting to revisit this decision and evaluate the cost of probe retries and how that affects accuracy. It would also be useful to better understand alternative points in the time/space sampling trade-off (Section 4).

Our study of firewalls in Section 6 focuses on visible firewalls, yet we know many firewalls are configured to simply consume unexpected packets. A deeper study of firewall usage, possibly using multiple, complementary detection approaches is likely to clarify trends in firewall deployment.

## 11 Conclusions

This paper is the first work, to our knowledge, to measure the population of hosts at Internet edge. We showed that censuses that walk the entire IPv4 address space, and surveys of about 1% of that space, provide complementary ways to evaluate the Internet. Our preliminary results discuss address space utilization, bounds on the number of servers, and deployment of visible firewalls. More importantly, we expect this methodology and our datasets to broaden the field of Internet measurements from routers and traffic to the edge.

## Acknowledgments

This material is based on work partially supported by the United States Department of Homeland Security contract number NBCHC040137 (”LANDER”). All conclusions of this work are those of the authors and do not necessarily reflect the views of DHS.

The authors would like to thank Tom Lehman and Forrest Houston at ISI East for hosting our probing machines there. We thank Los Nettos staff for their support of this work; help from Jim Pepin (now at Clemson), Maureen Dougherty, Walt Prue, and Sandy George, has been essential. We also thank ISI's Information Processing Center, particularly Christine Smiley and Drew Jongbloed, for handling first-round complaints about why we are pinging people.

## 12 References

[1] Mark Allman, Wesley M. Eddy, and Shawn Ostermann. Estimating loss rates with TCP. *ACM Performance Evaluation Review*, 31(3):12–24, December 2003.

[2] Internet Assigned Numbers Authority. Internet protocol v4 address space. web page `http://www.iana.org/assignments/ipv4-address-space`, September 2002.

[3] Internet Assigned Numbers Authority. ICMP type numbers. web page `http://www.iana.org/assignments/icmp-parameters`, March 2007.

[4] Rob Beck. Passive-aggressive resistance: OS fingerprint evasion. *The Linux Journal*, September 2001.

[5] T. Bu, L. Gao, and D. Towsley. On characterizing BGP routing table growth. Proceedings of IEEE GlobalInternet 2002, November 2002.

[6] Internet Software Consortium. Internet domain survey. web page `http://www.isc.org/ds`.

[7] S. Deering and R. Hinden. Internet protocol, IP version 6 specification. RFC 2460, Internet Request For Comments, December 1998.

[8] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. AS relationships: Inference and validation. *ACM Computer Communication Review*, 37(1):29–40, January 2007.

[9] Nick Duffield and Matthias Grossglauser. Trajectory sampling for direct traffic observation. In *Proceedings of the ACM SIGCOMM Conference*, pages 179–191, Stockholm, Sweeden, August 2000. ACM.

[10] Ed3f. Firewall spotting and networks analysis with a broken crc. `http://www.phrack.org/archives/60/p60-0x0c.txt`, December 2002.

[11] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the ACM SIGCOMM Conference*, pages 251–262, Cambridge, MA, USA, September 1999. ACM.

[12] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless interdomain routing (CIDR): an address assignment and aggregation strategy. RFC 1519, Internet Request For Comments, September 1993.

[13] Lixin Gao. On inferring automonous system relationships in the internet. *ACM/IEEE Transactions on Networking*, 9(6):733–745, December 2001.

[14] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proceedings of the IEEE Infocom*, Tel-Aviv, Israel, March 2000.

[15] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modelling, and analysis of a peer-to-peer file-sharing workload. In *Proceedings of the 19th Symposium on Operating Systems Principles*, pages 314–329, Bolton Landing, NY, USA, October 2003. ACM.

[16] T. Hain. A pragmatic report on IPv4 address space consumption. *The Internet Protocol Journal*, 8(3), 2004.

[17] A. S. Hedayat and Bikas K. Sinha. *Design and Inference in Finite Population Sampling*. John Wiley & Sons, Inc., 1991.

[18] B. Huffaker, D. Plummer, D. Moore, and K. C. Claffy. Topology Discovery by Active Probing. In *Proceedings of the Symposium on Applications and the Internet*, January 2002.

[19] Eddie Kohler, Jinyang Li, Vern Paxson, and Scott Shenker. Observed structure of addresses in IP traffic. In *Proceedings of the 2ndACM Internet Measurement Workshop*, pages 253–266, Marseille, France, November 2002. ACM.

[20] Craig Labovitz, Abha Ahuja, Abhijit Abose, and Farnam Jahanian. Delayed Internet routing convergence. In *Proceedings of the ACM SIGCOMM Conference*, pages 175–187, Stockholm, Sweeden, August 2000. ACM.

[21] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *ACM/IEEE Transactions on Networking*, 2(1):1–15, February 1994.

[22] Lun Li, David Alderson, Walter Willinger, and John Doyle. A first-principles approach to understanding the Internet's router-level topology. In *Proceedings of the ACM SIGCOMM Conference*, pages 3–14, Portland, Oregon, USA, August 2004. ACM.

[23] Xiaoqiao Meng, Zhiguo Xu, Beichuan Zhang, Geoff Huston, Songwu Lu, and Lixia Zhang. IPv4 address allocation and the BGP routing table evolution. *ACM Computer Communication Review*, 35(1):71–80, January 2005.

[24] Wolfgang Mühlbauer, Olaf Maennel, Steve Uhlig, Anja Feldmann, and Matthew Roughan. Building an AS-topology model that captures route diversity. In *Proceedings of the ACM SIGCOMM Conference*, pages 195–204, Pisa, Italy, September 2006. ACM.

[25] H. Narayan, R. Govindan, and G. Varghese. On the impact of routing and address allocation on the structure and implementation of routing tables. In *Proceedings of ACM SIGCOMM Symposium on Network Architectures and Protocols*, Karlsruhe, Germany, August 2003.

[26] NJABL. Not just another bogus list. `http://www.njabl.org/`, 2007.

[27] p0f Project. p0f passive os fingerprinting. `http://lcamtuf.coredump.cx/p0f.shtml`, September 2006.

[28] Vern Paxson. End-to-end Internet packet dynamics. *ACM/IEEE Transactions on Networking*, 7(3):277–292, June 1999.

[29] ANT Project. Ant project address space scanner. `http://www.isi.edu/ant/software/`, May 2007.

[30] ANT Project. Ant project datasets. `http://www.isi.edu/ant/traces/`, May 2007.

[31] NMAP Project. Nmap network security scanner. `http://www.insecure.org/nmap/`, 1997.

[32] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address allocation for private internets. RFC 1918, Internet Request For Comments, February 1996.

[33] C. E. Shannon. Communication in the presense of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21, January 1949.

[34] Yuval Shavitt and Eran Shir. Dimes: let the internet measure itself. *SIGCOMM Comput. Commun. Rev.*, 35(5):71–74, 2005.

[35] Matthew Smart, G. Robert Malan, and Farnam Jahanian. Defeating TCP/IP stack fingerprinting. In *Proceedings of the USENIX Security Symposium*, pages 229–240, Denver, Colorado, USA, August 2000. USENIX.

[36] F. Donelson Smith, Felix Hernandez, Kevin Jeffay, and David Ott. What TCP/IP protocol headers can tell us about the web. In *Proceedings of the ACM SIGMETRICS*, pages 245–256, Cambridge, MA, USA, June 2001. ACM.

[37] SORBS. Sorbs dynamic user and host list. `http://www.au.sorbs.net/faq/dul.shtml`, 2004.

[38] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, 2004.

[39] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings of the IEEE Infocom*, pages 618–627, New York, NY, USA, June 2002. IEEE.

[40] H. Tangmunarunkit, R. Govindan, S. Jamin, and S. Shenkerand W. Willinger. Network Topology Generators: Degree-Based vs. Structural. In *Proceedings of ACM SIGCOMM*, pages 188–195, Pittsburgh, PA, 2002.

[41] The PREDICT Program. Predict: Protected repository for the defense of infrastructure against cyber-threats. `http://www.predict.org`, January 2005.

[42] Paul F. Tsuchiya and Tony Eng. Extending the IP Internet through address reuse. *ACM Computer Communication Review*, 23(1):16–33, January 1993.

[43] Yinglian Xie, Fang Yu, Kannan Achan, Eliot Gillum, Moises Goldszmidt, and Ted Wobber. How dynamic are IP addresses? In *Proceedings of the ACM SIGCOMM Conference*, page to appear, Kyoto, Japan, August 2007. ACM.