Selecting Representative IP Addresses for Internet Topology Studies*

USC/ISI Technical Report ISI-TR-2010-666, June 2010

Xun Fan John Heidemann
USC/Information Sciences Institute

ABSTRACT

An *Internet hitlist* is a set of addresses that cover and can *represent* the the Internet as a whole. Hitlists have long been used in studies of Internet topology, reachability, and performance, serving as the destinations of traceroute or performance probes. Most early topology studies used manually generated lists of prominent addresses, but evolution and growth of the Internet make human maintenance untenable. Random selection scales to today's address space, but most random addresses fail to respond. In this paper we present what we believe is the first automatic generation of hitlists informed censuses of Internet addresses. We formalize the desirable characteristics of a hitlist: *reachability*, each representative responds to pings; *completeness*, they cover all the allocated IPv4 address space; and *stability*, list evolution is minimized when possible. We quantify the accuracy of our automatic hitlists, showing that only one-third of the Internet allows informed selection of representatives. Of informed representatives, 50–60% are likely to respond three months later, and we show that causes for non-responses are likely due to dynamic addressing (so no stable representative exists) or firewalls. In spite of these limitations, we show that the use of informed hitlists can add 1.7 million edge links (a 5% growth) to traceroute-based Internet topology studies Our hitlists are available free-of-charge and are in use by several other research projects.

1. INTRODUCTION

Smooth operation of the Internet is important to the global economy, so it is essential that Internet users, providers, and policy makers understand its performance and robustness. Although on the surface, individuals care only about their personal performance, a full diagnosis of "why is my web connection slow?" must consider not just the user's "first mile" connection, but dozens of servers that affect performance [8]. Web content providers invest great effort in optimizing page load times to sub-second values [22] and in building distributed content distribution networks that manage traffic (for example, [13]). Policy makers debate questions about universal access [32], a nation's relative availability for broadband access [23], and the robustness of what is recognized as critical infrastructure.

To answer these questions, network researchers, operations, and industry have developed a number of tools to map the Internet [16, 14, 25, 19, 21, 7], evaluate performance [31, 19, 22], consider questions about routing and reachability [30, 3], or the performance of replica placement (examples include [31, 9]), and evaluate topology robustness [1]. With the Internet's lack of centralization and multiple overlapping global "backbones", active probing plays an essential role in this process, with traceroute and ping and their variants providing the main source of router-level reachability. While one may add AS-level views [4], the Internet's router-level topology is the focus of this paper. Different router-level studies either target specific networks [25] or the whole Internet. Here we are most interested in observing the whole Internet—more than three billion allocated IPv4 addresses.

Studies of the entire Internet typically employ a *hitlist*—a list of IP address that can *represent* the billions of allocated addresses. The defining characteristic of a a hitlist is *completeness*, where a representative is chosen for every autonomous system or, in our case, for every allocated *block* of addresses defined by a /24 prefix, the smallest unit typically present in a default-free routing table. Representatives provide a 256-fold (or more) reduction in scanning size, allowing Internet-wide studies to take place in hours instead of months.

Although completeness is necessary to study the whole Internet, an ideally hitlist is also responsive and stable. A *responsive* representative replies to ICMP messages, allowing traceroute to confirm a path to the edge of the network, and ping to measure round-trip time to an edge host. To support longitudinal studies, the hitlist should be *stable*, with representative identities not changing frequently or arbitrarily.

Although hitlists are easy to define and have been used in topology studies for many years (we review related work in Section 2), they are surprisingly hard to create and maintain. Early hitlists were built manually from well known sites [19], but the size of the Internet and rate of churn in even well-known servers made manual maintenance unten-

*This work is partially supported by US DHS contract number NBCHC080035, and John Heidemann by NSF grant number CNS-0626696. The conclusions of this work are those of the authors and do not necessarily reflect the views of DHS or NSF.

able as it quickly became incomplete. More recent studies have typically used randomly chosen representatives. While randomness has some advantages (it can be statistically unbiased), it sacrifices secondary goals of stability and responsiveness.

The contribution of this paper is to provide a new, automated method of hitlist generation that provides complete coverage while maximizing stability and responsiveness.[1] Our hitlists are constructed (Section 3) by mining data from IP address censuses, complete, ping-based enumerations of the allocated IPv4 address space taken every two to three months [17].

The second contribution of our work is to evaluate our hitlists (Section 4). Our hitlists are 100% complete as of when they are constructed, although when we have no history (in about two-thirds of the blocks) we select representatives at random. We define the *accuracy* of our hitlists has how many representatives are responsive three months after the hitlist is taken. We find that two-thirds of the allocated address space never responds to ICMP probes and so never has responsive representatives. Of the remaining, responsive Internet, our hitlists select representatives that are responsive about 55% of the time. To our knowledge we are the first to study hitlist effectiveness and accuracy.

The final contribution of our work is what hitlists reveal about that nature of the Internet itself. We were surprised that, in spite of such complete input data, the responsiveness of our predicted representatives is not higher. We believe this upper bound on productiveness characterizes the portion of the Internet that has an inherently high rate of address churn. One corollary of this limit to representative responsiveness is that no manual system could ever have been successful due to natural turnover of addresses in parts of the network. We also characterize the distribution of addresses in each block and show that it strongly reflects address allocation patterns (Section 5).

We make our hitlists available free-of-charge, and they are already being used by several research projects. In Section 6 we discuss the security and policy issues involved in sharing this data.

2. RELATED WORK

Hitlists are used in active probing for studies of topology [16, 14, 18, 25, 29, 21, 24, 7], performance [31, 19, 22], and reachability [30, 4]. and for other purposes [31, 1, 9]. Each of these studies uses some hitlist (sometimes called a seed or probe list) generated manually, randomly, or automated from several sources. We review each hitlist generation method next.

Early topology work used manually generated lists. Skitter is a well-known measurement tool developed at CAIDA, to study the router-level Internet topology [18]. It uses traceroutes from multiple locations to a hitlist of destinations. Their target address list was manually built from many sources, including tcpdump from the UCSD–CERF link, hostnames from search engines, and intermediate addresses seen from their own traces records. In 2000 their hitlist included about 313,000 destinations, and by 2004 it had grown to 971,080. While their hitlist was of high quality, they found it very labor-intensive to maintain, and responsiveness degraded over time as destinations changed. (They report a 2–3% loss for their initial web-server based list [18].) The cost of manual list maintenance prompted them to change to random probing with Archipelago. More recently, Maennel has maintained a manual list, derived from the Skitter list, but augmented with guided scanning to cover each AS and provide 306,708 representatives. They require reachable addresses study routing reachability [4]. We used a version of their list to seed our initial stable list, but our techniques provide much greater coverage at lower cost. Unlike all of these manual hitlists, our goal is to fully automate hitlist generation to allowing more complete and timely coverage.

Random representative selection allows low-cost generation of hitlists to much larger numbers of networks. Mercator developed *informed random probing* to adaptively adjust its probe list based on prior results [16]. By adaptively growing the hitlist, Mercator strives to quickly and efficiently discover a topology while minimizing hitlist size. Archipelago (Ark) is a measurement platform designed to support traceroute and other measurements [7], effectively a next-generation Skitter. Ark's hitlist covers all routed /24 blocks, choosing a random last-octet within each /24 block, The random hitlists in Mercator and Ark are essential to cover the millions of /24 networks in today's Internet, but Mercator's adaptive algorithm means completeness is uncertain (although efficiency, not completeness, was their goal), and random probing in both Mercator and Ark sacrifice responsiveness of the destination address. Our hitlist also provides complete coverage, but it also maximized responsiveness. In Section 4.4 we evaluate the degree to which informed hitlist generation may improve topology discovery.

Rather than a random destination, DisCarte's hitlist selects the .1 address in each /24 block of the routed address space. DisCarte [24] adds record route information to traceroute probing to obtain more accurate and complete network topologies. They require a responsive destination, and find 376,408 responsive representatives in the .1 address of each routed /24. Our work confirms that the .1 address is responsive twice as often as the median address (Section 5), but we suggest that census-informed representative selection can get much better responsiveness.

Finally, there has been some work in IPv6 topology discovery. The Atlas system uses a manually generated list built from 6bone destinations [29], then expanded based on discoveries. Our approaches use a full address-space census that applies only to the IPv4 address space, so combinations of active and passive methods as proposed in Atlas are essential for IPv6. As future work, comparison of our active of in IPv4 against a passive hitlist provide a basis for inferring coverage in IPv6.

3. METHODOLOGY

We next describe the requirements of an IP hitlist (Section 3.1), and how we transform census data (reviewed in Section 3.2) using several possible prediction methods (Section 3.3) to get a good quality hitlist. We also provide some details on how our implementation copes with Internet-sized datasets.

3.1 Hitlist Requirements

Our goal is to provide representatives that are responsive,

[1]We would like to thank Randy Bush for suggesting the idea that our address censuses data could support hitlist generation.

complete and stable.

By *responsive*, we mean each representative is likely to respond to an echo request with an echo reply instead of an ICMP error code. As we describe below, select representatives that have responded frequently in the past. We do not guarantee that that address responded in the most recent census, but we bias our selection to favor recent results. We consider several prediction functions below in Section 3.3.

By *complete*, we mean we report one representative address for every allocated /24 block. Some groups have used other definitions of completeness, such as one representative per AS, or per routed BGP block. AS- or BGP-complete hitlists will be both sparser and smaller than /24-complete maps, since ASes typically include routes for many prefixes, and routed prefixes often cover blocks larger than /24s. However, we select /24 blocks so that the hitlist is decoupled from the routing system, since routes differ depending on when and where they are taken.

By *stable*, we mean and that representatives do not change arbitrarily. We change representatives when a new representative would significantly improve the score for that block, typically because a representative has ceased to be reachable. *Inertia* provides a bias to avoid changing representatives. Currently switch addresses when they improve the score significantly (inertia is 0.34); in Section 4.3 we examine the inertia threshold is set and how it affects accuracy.

These goals can be in conflict. For example, completeness requires that we select representatives that may be non-responsive. To guarantee representatives for all allocated addresses, we select representatives even for blocks that have no recent responses. We also select representative for blocks that have never responded. In both cases, we annotate these representatives with distinguished scores.

3.2 Background: Internet censuses

Our main goal with a hitlist is to predict the future: a representative should be responsive in the future. Our tool to make this prediction is data from past responses. Hitlists selection leverages Internet censuses that have been taken regularly since 2003 [17].

Each Internet census is the results of a ping (an ICMP ECHO REQUEST message) sent to every allocated IPv4 address. Censuses are far from perfect: a census must be taken carefully to avoid ICMP rate limiting or transient router errors, and firewalls reduce ping response rates by around 40%. Hitlist, however, prefer hosts that are ICMP responsive, since traceroute consists of iterated, TTL-limited ICMP messages. Firewall-limited censuses are therefore ideal for hitlist generation.

It takes 2–3 months to carry out a full census (the IPv4 space has more than 3 billion allocated unicast addresses). For this paper we consider censuses starting in Mar 2006 as shown in Table 1, since censuses before this date used a slightly different collection methodology. The results of this paper use all 22 censuses taken over the four years preceding analysis, but we expect to update our results as new censuses become available.

A census elicits a number of responses, including ECHO REPLY messages as well as a variety of errors. Each census is quite large, and more than 3 billion records per census, 22 censuses is over 260GB of raw data. We therefore pre-process all censuses into a *history map* convenient for analysis. A history map consists of a bitstring for each IP address where each 1 indicates a positive response, and a 0 indicates either a non-response or negative response. In paper, we only consider echo replies ("positive" responses) as indicating a responsive address. We next show how this history map can predict future response rates.

Censuses	Date	Duration(days)
it11w	2006-03-07	23
it12w	2006-04-13	24
it13w	2006-06-16	31
it14w	2006-09-14	31
it15w	2006-11-08	61
it16w	2007-02-14	50
it17w	2007-05-29	52
it18w	2007-09-14	47
it19w	2007-12-18	48
it20w	2008-02-29	86
it21w	2008-06-17	49
it22w	2008-09-11	35
it23w	2008-11-25	29
it24w	2009-02-03	29
it25w	2009-03-19	29
it26w	2009-05-27	31
it27w	2009-07-27	25
it28w	2009-09-14	30
it29w	2009-11-02	30
it30w	2009-12-23	29
it31w	2010-02-08	30
it32w	2010-03-29	29

Table 1: IPv4 censuses [28] used in this paper.

3.3 Prediction Method

Of our hitlist goals of responsiveness, completeness, and stability, completeness and stability are under our control, but responsiveness requires predicting the future. Our guidance in this task is the prior history of each address. We next review several *prediction functions* that strive to select the best representative for each /24 block, where best is most likely to respond in the future.

Prediction functions take the prior history of address a as input and weights that history in different ways. History for a is identified as $h_i(a)$, numbered from 0 (oldest) to $N_h - 1$, the most recent observation. We consider several different weights $w(i)$ to get the scores $s(a)$ in the form:

$$s(a) = \sum_{i=0}^{N_h-1} r_i(a)w(i)$$

$r_i(a)$ is the response of address a to the ith probe. For each block of addresses, the address with the highest $s(a)$ is selected as the best representative. We may bias this by prior representatives to promote stability. In the case of ties and no prior representative we select any top scoring address in the block at random.

We considered several possible weights $w(i)$. The simplest is $w(i) = 1$, so all responses are *averaged*. To give more recent observations greater influence we consider two biased weights. With *linear* weighting, $w(i) = (i + 1) * 1/N_h$, and for a *power* function, $w(i) = \frac{1}{N_h - i}$. Weighting of each observation for an 8-observation history is shown in Figure 1.

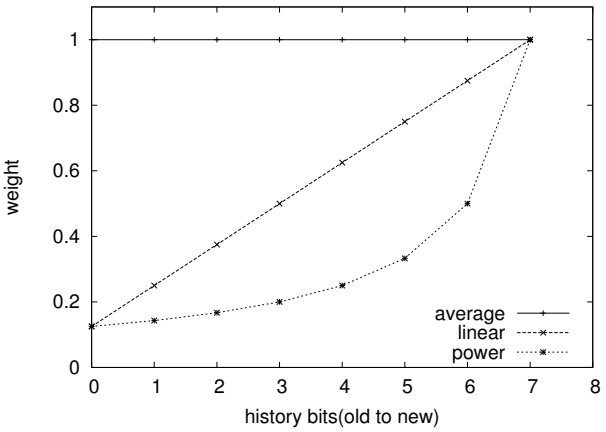


Figure 1: bits weight for different function

In addition, we can normalize scores by to the maximum possible score (the minimum in all cases is zero), allowing all to fall in the range 0 to 1.

As an example of the different functions, Figure 2 shows scores for three different weights and different history lengths. For simplicity, we assume $N_h = 8$, shorter than we use in practice (in Section 4.1.2 we vary history duration). We consider three cases, all with 4 of 8 responding, but either responding most recently (Figure 2a), in the middle past (Figure 2b), or alternating response and non-response (Figure 2c). To a first approximation, all three weights are about the same, particularly with intermittent responsiveness in Figure 2c. The differences in decay rates are more obvious when responsiveness is consistent for blocks of time, with power and linear decay faster than average in Figures 2a and 2b. Finally, difference in history duration make a large difference when a block is non-responsive, comparing the left and right parts of Figures 2a and 2b, and these effects are even greater when comparing across weights (for example, compare history durations 1–4 of Figures 2a and 2b).

This framework provides flexibility, but requires setting several parameters. We later evaluate which weighting is best (Section 4.1.1), how much history is beneficial (Section 4.1.2), and the underlying reasons addresses are difficult to predict (Section 4.1.3).

3.4 Gone-Dark Blocks

Firewalls are probably the greatest impediment to active problem, since a conservative firewall can suddenly stop traffic an entire block. We will see in Section 4.1.3 that *gone dark* blocks are one cause of poor representative responsiveness. A gone-dark blocks is one that contained responsive addresses for some period of time, but then becomes unresponsive and stays that way, due to firewall or possibly renumbering. While we must select a representative for each allocated block, even if populated only by non-responsive addresses, we would like to indicate our low expectations for gone-dark blocks.

We define a block as *gone dark within history* N_d if, for the most recent N_d observations, no address in the block responded, even though we had some positive response before N_d observations.

We add gone-dark analysis to our hitlist generation by overriding the representative's score with a designated "gone-dark" value to indicate our skepticism that it will reply. We explored different values of N_d and ultimately select $N_d = N_h = 16$, identifying only those addresses whose responses have aged-out of our history as gone-dark. We use this large value of N_d because this value maximizes the absolute number of responsive representatives, while only decreasing the percentage of responsive, predicted representatives a small amount.

For gone-dark blocks, we still select the representative as the address with the best score. For allocated but never-responsive blocks, we select the .1 address as the representative because that is most likely to be first used (Section 5). In Section 4.1.3 we show the contribution of gone-dark blocks to responsiveness.

3.5 Hitlist Description

To summarize, our hitlist contain three kind of representatives for all allocated /24 blocks: informed and predicted representatives, where we select the best responder; gone-dark representatives, where some address once responded but has not recently; and allocated but never-responsive blocks, where we pick .1 as the representative.

Table 6 lists the hitlists we have publicly released to-date. We identify hitlists by the name of the last census used in their creation, and include the number of censuses in the history. Thus HL28/16 uses 16 hitlists through it28w. When necessary, we add the gone-dark window, so HL28/16-3 uses a window of 3. If no gone-dark window is specified, we disable gone-dark processing.

In addition to these public hitlists, Tables 2 and 3 show unreleased hitlists used to evaluate our methods.

3.6 Implementation

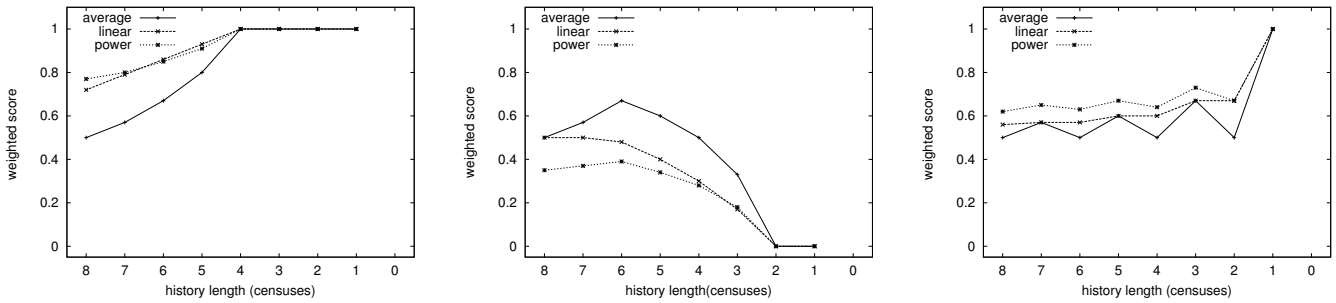
Analysis of a four-year history of the entire IPv4 address space is quite data intensive. We next briefly describe our analytic approach to assist others considering similar evaluation. The main challenge is dataset size. The full size of a raw census is about 12GB, so an 22-census history is over 260GB of data. Even when reduced to a simple responsive bitmap, a single census is 2^{32} bits (16MB) in size, with the full history about 32GB.

Fortunately, our analysis parallelizes easily—all analysis is done on history for /24 blocks, and there are no inter-block dependencies. We employ the Hadoop implementation [10] of map/reduce [11] to parallelize computation over a cluster of about 40 computers with about 120 CPU cores. With this parallelism, join in a new census into an existing history takes about half an hour, and evaluation of a new hitlist takes about another half hour. (our code is written in Perl and not optimized for speed). Our map function groups results by block, while the reducer carries out the join or evaluation.

4. EVALUATION

We next evaluate the success of our hitlist: how accurate are its predictions and how complete and stable is it? We first consider how responsiveness is affected by choices in our prediction mechanism. In Section 4.1.3 then look at causes of prediction failure. Finally, we consider completeness and stability.

4.1 Responsiveness



(a) 0000 1111. (b) 0011 1100. (c) 0101 0101.

Figure 2: Comparison of three history functions for selected addresses.

Our primary goal with prediction is *responsiveness*: how accuracy is our prediction that the representatives in a hitlist will respond in the future? We can define based on the number responding in the future, N_r, from the number of predicted representatives (including representatives of gone-dark and informed predicted blocks) N_p as:

$$\alpha = \frac{N_r}{N_p}$$

Responsiveness accuracy is affected by our choice of history weighting and length. We consider these next, and then consider structural reasons perfect accuracy is impossible to achieve.

Our general approach to test responsiveness is to generate a hitlist, then evaluate it against ICMP probes in the next census. For example, the first line of Table 2 evaluates HL19/8, generated from the eight censuses from it12w through it19w, tested against it20w. This approach has the advantage of supporting retroactive evaluation of hitlist quality under different, controlled conditions. However, it also means each representative is only given one opportunity to be available. For this reason we report exact counts of results, without error estimates such as standard deviation. We evaluate repeatability of our results by considering multiple hitlists at different times.

4.1.1 Comparing History Weights

We first consider how our weighting of prior history affects accuracy. Here we assume a history duration of 8 prior censuses (a reasonable choice as evaluated next in Section 4.1.2), and from that history we predict the results of the next census for the three weights we defined in Section 3.3. Since the network is dynamic, our expectation is that biased weightings will perform best since they favor recent information over older information.

To answer this question, Table 2 compares our three weightings for several predictions. Each line evaluates a different hitlist as generated with three different weights, and evaluated for all predicted representatives (N_p). The most important observation is that *all weights* provide quite similar performance—the worst case responsiveness is only 5% worse than the best. Linear and power functions provide marginally better responsiveness. The examples of the weights in Figure 2 suggests on reason the difference is so small. For many histories, all three weights produce roughly the same relative scores.

	weighting function		
hitlist	**average**	**linear**	**power**
HL19/8	0.50	0.51	0.51
HL21/8	0.53	0.54	0.55
HL23/8	0.53	0.54	0.54
HL25/8	0.53	0.54	0.54
HL27/8	0.54	0.55	0.55

Table 2: Fraction of responsive representatives across 5 different hitlists for three different history weights.

4.1.2 Effects of History Duration

A second factor that can affect responsiveness is the duration of history considered in a prediction. Does more history provide more information, or does very old information become irrelevant or even misleading?

To study this question, we considered all history available to us at time of analysis—then we had 18 Internet censuses covering 3.5 years. We consider only the power weighting of history, and look at the responsiveness of our predictions.

Table 3 shows responsiveness of our predictions as a function of history length, for five predictions. We see that very short histories are insufficient: prediction rates are a 1–2% lower when fewer than 8 (about 1.5 years) observations are considered. On the other hand, we see no difference in prediction accuracy for histories from 8 to 16 censuses. (We also looked at history duration with the average function, and found there that long histories became slightly less accurate, although only by 1–2%. This observation argues in favor of a weighting that decays by history, like power.)

Finally, while longer histories may not improve the fraction that respond, it does provide information that allows *more* representatives to be selected. Table 3 shows the absolute number of responders as a function of history duration. Longer history allows 20k more responders with length 16 than with length 8. More history always increases the number responding, although with diminishing returns past a 12 censuses or so.

In practice, the incremental cost of longer history lengths is not large. So we use a history length of 16 censuses in our production lists.

Although 8 censuses provides slightly betters results, the faction responding, only 55%, seems lower than we might ex-

hitlist	predicted representatives (N_p)	Responsive representatives and fraction 4	8	12	16
HL19/-	3,091,646 (100%)	1,558,620 (50%)	1,586,303 (51%)	—	—
HL21/-	3,386,540 (100%)	1,813,276 (54%)	1,846,019 (55%)	—	—
HL23/-	3,613,523 (100%)	1,925,322 (53%)	1,948,634 (54%)	1,950,960 (54%)	—
HL25/-	3,794,973 (100%)	2,007,138 (53%)	2,049,607 (54%)	2,059,019 (54%)	—
HL27/-	3,971,208 (100%)	2,135,337 (54%)	2,179,777 (55%)	2,193,062 (55%)	2,200,674 (55%)

Table 3: Responsive representatives with power weighting across 5 different hitlists for different history length.

pect. We therefore next consider causes of non-responsiveness.

4.1.3 Causes of Failed Responses

We found the observation that our best methods get only 55% responsiveness seems somewhat surprising. Surely such a large amount of history (over three years of full censuses) can be explored somehow to select representatives with greater accuracy. To answer that question, we next explore the causes of why representatives fail to respond. Our conclusion is that *it is unlikely that any prediction can do better than about 70%* because of the use of dynamic address assignment and firewalls.

To support this claim, Table 4 counts prediction failures for HL28/16, tested against it29w (We found roughly similar results in examination of HL31/16 evaluated against it32w.) We see that 44% of representatives are non-responsive (1.8M of the 4M blocks). Two explanations account for the majority of our misses: blocks that use only dynamic address assignment, and "gone-dark" blocks. We consider each of these below.

While dynamic addressing and firewalls are target-specific causes of representative non-responsiveness, measurement error is a possible source of uncertainty. We believe that Internet census-taking methodology reduces these sources of error to random noise for reasons described in prior work [17]. To summarize briefly: we monitor the network hosting the probes for local routing outages. Probes are in pseudorandom order, so routing outages in the middle or near the destination result in lower responsiveness in proportion to outage rates, but randomly distributed. Pseudorandom probing is spread over two months, so the probe rate to any individual /24 is well below typical ICMP rate limits. We considered packet loss and routing outages in the middle or of the network or near probe sources are potential sources of error. For more complete discussion of sources of error in Internet census-taking, and validation studies, we refer to prior work [17].

Defining stable blocks: Blocks that lack stable addresses makes representative selection inherently difficult. In a block with a stable representative, it will likely remain responsive, but if all addresses in the block are unstable then the probability a representative will respond is equal to the occupancy of that block and independent of prior history. Addresses can lack stability either because the hosts using the addresses are only on intermittently, or because addresses in the block are allocated dynamically to a changing population of computers. Multiple groups have used different techniques to identify dynamically assigned addresses in the Internet [26, 33, 5]. A recent study estimates that about 40% of responsive Internet blocks are dynamic based on Internet address surveys using ICMP probes taken every 11 minutes for two weeks [5]. (We assume here that non-stable blocks are primarily due to dynamic addressing.)

To evaluate the prevalence of stable and non-stable blocks, we would like to identify them from the history that we collect. Prior analysis of surveys used address *availability* and *volatility* to identify dynamic addressing. Availability is the fraction of times the address responds in all probes, while volatility is the fraction of times the address changes between responsive and non-responsive [5]. While appropriate for survey data with 11-minute probes, volatility makes less sense when probes are months apart.

To identify stable blocks with infrequent probes, define a new metric, *truncated availability*, the fraction of time an address responds from its first positive response. More formally, if $r_i(a)$ is the response of address a to the ith probe, the raw and scaled availability, $A^*(a)$ and $A(a)$ (from [5]) and truncated availability, $A^t(a)$ are:

$$\begin{aligned} A^*(a) &= \sum_{1}^{N_h} r_i \\ A(a) &= A^*(a)/N_h \\ A^t(a) &= A^*(a)/L^*(a) \end{aligned}$$

where $L^*(a)$ is the length of a history, in observations, from the first positive response to the present.

While both volatility and truncated availability are correlated, we found that low volatility and high truncated availability are both good predictors a stable block. Low A^t values are a good predictor of intermittently used addresses. Continuing the examples in Figure 2, 00001111 has $A^t = 1$, while 01010101 has $A^t = 0.57$.

While A^t is good at differentiating between these solid (00001111) and intermittent (01010101) addresses, it interacts with gone-dark addresses, which will have a string of trailing 0s. We therefore consider A^t only in concert with A^*, the absolute number of positive responses. Small A^* values (say, less than 5) indicate addresses that were only briefly responsive or are quite new.

From these, we define a stable representative as $A^t \geq 0.9$, however we look carefully at representatives where $A^* < 5$. We find that 51% of all representatives are not stable by $A^t \geq 0.9$, some higher than other independent observations 40% [5], and of 34–61% [33] (these values are for a random sample of DNS and for Hotmail users, respectively). However, more than half of these were only observed briefly ($A^* < 5$). We do not claim strong validation of this exact percentage because each work is a percentage of a different populations, and different definitions of what is dynamic or not stable. We claim only that our metric is in the right or-

	HL28/16			HL31/16		
predicted representatives (N_p)	4,055,193	100%		4,307,644	100%	
not stable	2,183,353	53%		2,276,207	52%	
gone dark	703,987	17%		772,014	18%	
responsive (N_r)	2,250,091	56%		2,560,420	59%	
non-responsive (N_n)	1,805,102	44%	[100%]	1,747,224	41%	[100%]
non-responsive and not stable (only)	805,136	20%	[45%]	786,881	18%	[45%]
non-responsive and gone dark only	0	0%	[0%]	0	0%	[0%]
non-responsive, not stable and gone-dark	693,832	17%	[38%]	766,338	18%	[44%]
non-responsive, just unlucky	306,134	7%	[17%]	194,005	5%	[11%]

Table 4: Causes of unsuccessful representatives predicted from HL28/16 and HL31/16, evaluated against responses in it29w and it32w. We don't apply gone-dark window on prediction here, gone-dark blocks are detected separately with gone-dark window size of 3.

	$A^* < 5$	$A^* \geq 5$	sum of A^t
$A^t < 0.9$	700,328 (39%)	583,976 (32%)	1284304 (71%)
$A^t \geq 0.9$	214,664 (12%)	306,134 (17%)	520798 (29%)
sum of A^*	914,992 (51%)	890,110 (49%)	1,805,102 (100%)

Table 5: Fraction of representatives that are non-responsive, based on combinations of A^t and A^* (HL28/16 tested against it29w).

der of magnitude and so provides some insight into sources of non-responsive representatives.

Re-evaluating causes of non-responsive representatives: With these definitions, we return to Table 4. We see that both gone-dark and not-stable blocks contribute three-quarters of our misses. Almost two-thirds are in not-stable blocks, with almost 40% gone-dark, and 27% of those overlapping as both gone-dark and not stable. We therefore claim that three-quarters of our non-responses are due either to new firewalls or selection of representatives in not-stable blocks, neither of which can *ever* have always responsive representatives.

To support the claim that lower A^t values correlate with poorer response, Table 5 breaks out the 1.8M non-responsive representatives by all combinations of A^t and A^*. We see that only 17% of non-responses come from stable blocks ($A^t > 0.9 \wedge A^* \geq 5$). Representatives with poor truncated availability ($A^t < 0.9$) account for more than two-thirds of non-responses, although many lower A^t values are limited by short histories ($A^* < 5$). We conclude there are many unstable blocks, such blocks simply cannot be expected to support stable representatives.

To show our choice of threshold for A^t does not alter our conclusion, Figure 3 shows the cumulative distribution of A^t for both non-responsive and responsive representatives. It shows a large difference in responsiveness for any value of A^t.

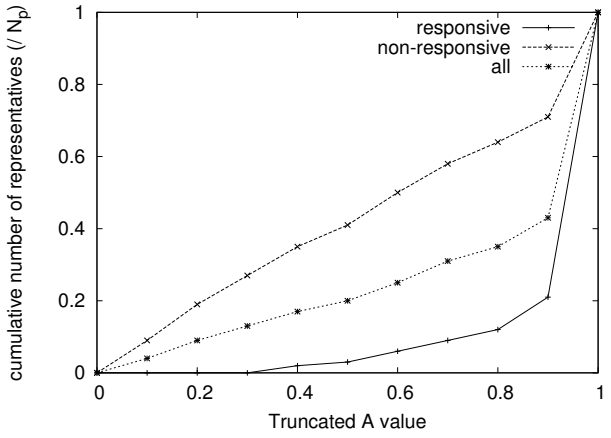


Figure 3: A^t cumulative distribution on N_p

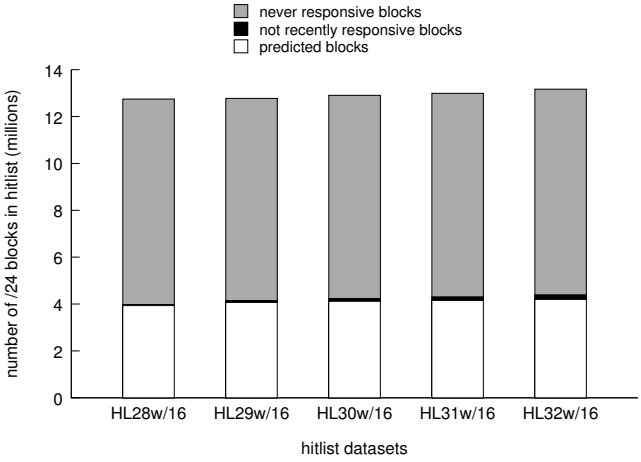


Figure 4: Relative size of hitlist components.

4.2 Completeness

To evaluate completeness, Figure 4 shows the absolute number of representatives for using 16-deep histories through five different censuses, and Table 6 shows the raw data. We consistently see that about one-third of blocks have some history data allowing an informed selection of representatives (the white region of the graphs, with around 4.2M blocks). By contrast, about two-thirds of blocks have never responded (the top grey regions)

In addition, this data shows gone-dark selection from Section 3.4. We identify about 0.3–1.5% of allocated blocks as formerly responsive (the black region in the middle of Fig-

class	HL28/16	HL29/16	HL30/16	HL31/16	HL32/16
allocated /24 blocks	12,774,056	12,774,056	12,905,128	13,036,541	13,167,613
never responding blocks	8,802,845	8,631,417	86,797,99	8,728,897	8,775,398
predicted blocks	3,971,211	4,142,639	4,225,329	4,307,644	4,392,215
gone-dark blocks	35,623	75,714	109,099	154180	195,216
informed prediction blocks	3,935,588	4,066,925	4,116,230	4,153,464	4,196,999
changed representatives	—	218,419	341,765	292,079	306,588
new representatives	—	171,428	82,690	82,315	84,571
responsiveness	2,200,993	2,344,539	2,411,662	2,451,351	

Table 6: Released hitlists to-date, by last census used in prediction (top). The top group of rows show hitlist composition, including churn (changed) and new representatives relative to the prior hitlist. The bottom line, responsiveness, evaluates the hitlist against the census.

ure 4).

To guarantee completeness, we select random representatives for never-responsive blocks. However, we can see that we can provide informed choices for only a third of blocks. Finally, we note that IANA only releases new allocation maps quarterly, and routing studies suggest this space becomes routable gradually [3], so we expect our hitlist to be useful for at least three months, about the frequency we update them.

4.3 Stability and Inertia

We next consider two aspects of hitlist stability: how much churn is there in the hitlist, with and without a representative inertia, and how much does inertia reduce prediction accuracy.

Recall that inertia is the amount I by which prediction score must improve to change representatives. An inertia $I = 0$ means we always pick the highest rank address in a block as the representative, independent of the representative in a hitlist based on a prior censuses. As inertia approaches 1, we will never switch representatives once chosen. For our production hitlists, we use $I = 0.34$.

Inertia on churn: We first consider how much inertia affects churn. Churn is that rate at which we switch representatives for established blocks. Table 6 shows the amount of churn for four hitlists when using our standard inertia $I = 0.34$. We see that the rate of churn is relatively constant with 5–7% of all informed predictions changing each census.

While Table 6 shows churn over time for a fixed inertia, in Figure 5 we vary inertia to observe its effect on churn. To estimate the relationship shown in this figure, we generate HL28/16, then modify it three times with censuses it29w, it30w, and it31w, with different levels of inertia. (Here we suspend gone-dark processing to focus only on inertia.) We then evaluate the hitlist against observations from census it32w. We evaluate inertia over several steps for two reasons. First, hitlist staleness is partially a function of time. Second, large values of inertia suppress changes in single or a few censuses.

As expected, Figure 5 shows that higher inertia suppresses churn, because it takes several new negative responses for a representative's score to change. In fact, weight selection means score can change only by 0.3 from one new census, and decrease to 0.5 from two new censuses since the weight decrease in our pow weighting, so with three new observations here, an inertia of 0.2 has one observations that might cause change, while $I = 0.4$ has two; $I = 0.6$, three; and $I = 0.8$ requires more than eight observations to change.

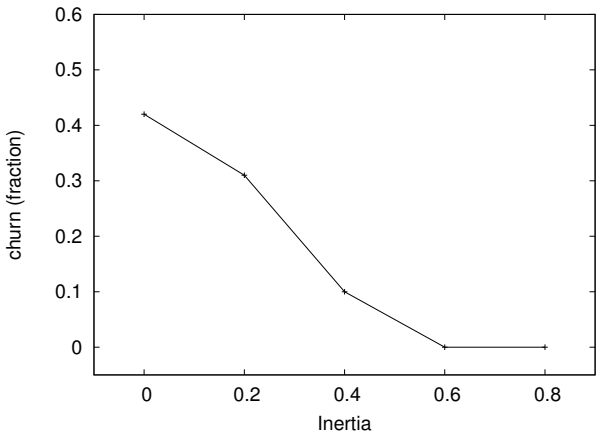


Figure 5: Effects of different inertia on representative churn (HL28/16; modified by it29w, it30w, and it31w; then tested against it32w).

Inertia on responsiveness: Inertia is selected to keep hitlists stable, reducing the amount of arbitrary representative turnover in long-running experiments. Such turnover can be eliminated by simply never changing representatives (setting $I = 1$), but prior experience shows that the responsiveness of a static hitlist will degrade over time as servers move, losing as much as 2–3% per month for the early Skitter web-server-based list [18]. We would therefore like to know the trade-off between inertia and representative responsiveness.

Figure 6 shows hitlist responsiveness for different values of inertia after this process. (This analysis was generated with the same multi-step process as Figure 5 described above.) We see that responsiveness degrades slightly for high inertia values, from 59% responsiveness with no inertia, to a low of 53% responsiveness when $I = 0.8$, when there are effectively no changes. We conclude that a moderate inertia has little effect on responsiveness costing at most 6% responsiveness, even over eight months.

4.4 Effects on Other Research

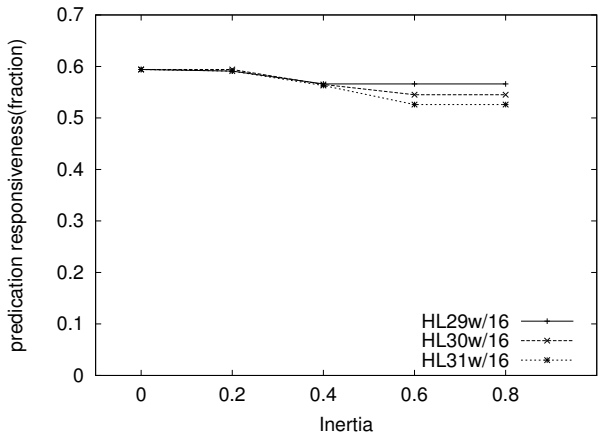


Figure 6: Effects of different inertia on responsiveness (HL28/16; modified by it29w, it30w, and it31w; then tested against it32w).

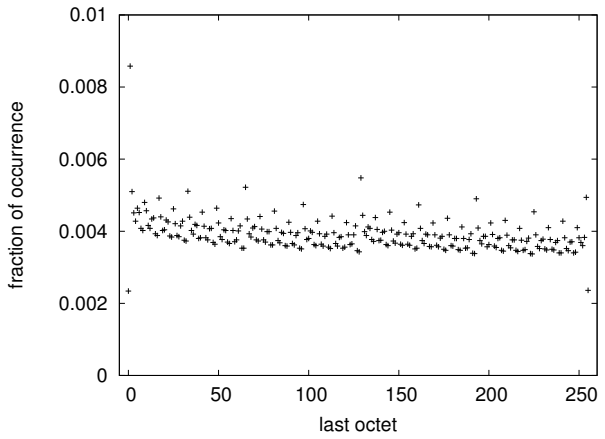


Figure 7: Frequency of responsiveness by last octet of the address (from it29w).

The above sections evaluate hitlists based on our goals: responsiveness, completeness, and stability. But hitlists are a tool to enable other research, so their ultimate benefits come by how they improve the quality of other network performance and topology studies.

One group of network performance studies *require* reachability in their destinations. These studies include those that evaluate performance [31, 19, 22], consider questions about routing and reachability [30, 3], or the performance of replica placement (examples include [31, 9]). Because these studies require end-to-end latency measurements, our representative selection methods optimize reachability within the constraints of sparse measurement. Our work also suggests directions for potential improvements: more frequent measurement could potentially better track reachable addresses in dynamically assigned blocks. In addition, our approach to stability assists evaluation of long-term performance trends.

Reachability is helpful but not essential for many topology studies (such as [16, 14, 25, 19, 21, 7]). Most topology studies employ traceroutes to study *paths* across the Internet. A traceroute attempts to discover all routers on the path towards a destination, but the presence or absence of the destination itself affects only the last hop. Difficulty in maintaining a hitlist, and the recognition that responsive targets are not essential prompted CAIDA to shift from a manually maintained hitlist [19] to random edge directions [7].

Although reachability is not essential for topology studies that focus on the core of the Internet, it is important for studies that wish to explore the *edge* of the network. We can get a rough estimate on the number of edge links that are missed by randomly selected representatives: about 4–7% of the Internet responds to ICMP probes [17], so we expect that 93% of random representatives do not respond. If 55% of our hitlists respond, that will improve edge detection for 48% of blocks. With 1.3 million allocated /24 blocks (as of it29), responsive hitlists will detect about *630,000 additional* links more than a random hitlist. By comparison, the core represents 33M links, so this increases the size of the discovered Internet topology by 2%. This simple analysis ignores correlation in the data, so it is only approximate.

To confirm this simple analysis, Table 7 directly compares the ITDK-2010-01 dataset [6] (from data taken Dec. 2009) with our HL29/16 (using censuses from 2007 through Dec. 2009). For ITDK, we consider a representative responsive if the traceroute reaches its destination. For our hitlist, we test against census it30w (finished in Jan. 2010). We then compare, for the /24 blocks in *both* ITDK and the predicted portion of our hitlist (N_p), the responsiveness of either method in each block. This comparison shows that informed representatives are 3.4× more responsive, and that a traceroute study that uses our hitlists would find about *1.7 million additional* edge links, a 5% additional coverage.

5. OTHER OBSERVATIONS

Given no knowledge about a /24 block, which address is most likely to be responsive?[2] This question has some bearing on which representative we should select for gone-dark blocks, or for newly-allocated blocks with no census data yet.

Discussions with network operators suggest some network practices are common. Often addresses are allocated sequentially from the start of an block, and network managers often use the first or last address in a block for the routers. Since address blocks are allocated on powers-of-two according to CIDR [15], we expect to see uneven use of the address space. Recent work has confirmed visibility of allocation blocks in census data [5], but not last-octet usage.

To evaluate this question, Figure 7 shows the distribution of responsiveness for the last octet for all in it29w. (We got similar results on it28w and it30w censuses.) Consistent with expectation, the most responsive octet is .1, responding 0.86% of the time, more than twice as often as the median responsiveness (0.38%), and 1.5× more frequent than .129 (0.55%), the next most responsive last octet.

Figure 7 shows a pattern in responsiveness, with responses being most frequent at addresses that are one greater than a power of two. The top ten are ranked .1, .129, .65, .33, .2, .254, .17, .193, .97, .9, and of these only 2 and 254 do not follow this pattern. To show this trend more clearly,

[2]We thank Kim Claffy for suggesting the question of last octet distribution for study.

Datasets	/24 blocks	responsive in ITDK	responsive in N_p
ITDK-2010	8,248,027	730,496	2,454,500
informed predictions through HL29/16 (N_p)	4,142,639	725,930	2,463,824
subnet-level intersection of both	4,008,861	725,930	2,454,500

Table 7: statistics on CAIDA and our hitlist

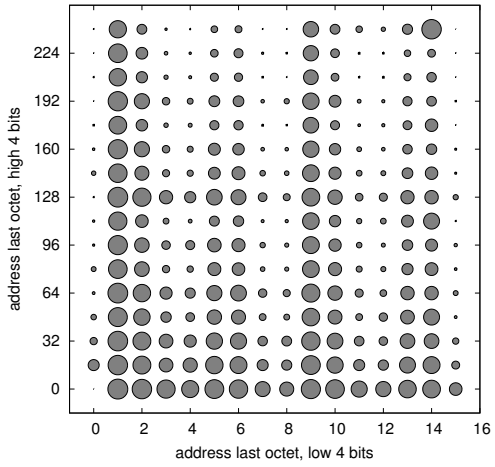


Figure 8: Rank (shown by circle area) of responsiveness by last octet, in a 16×16 grid by address (from it29w).

Figure 8 shows the rank of each last octet as the area of a circle, with the octets arranged in a sequential grid (so the x axis lists octets sequentially in groups of 16, while each step up the y axis is 16 more than the previous). The vertical lines correspond to more frequent responses, with $x = 1$ showing strong response from .1, .129, .65, etc., and $x = 9$ for .9, etc. The other prominent features are .254 and across $y = 0$ (.1, .2, .3, etc.). While showing ranks exaggerate what can be small absolute differences, these strong patterns show power-of-two allocations affect responsiveness.

6. SHARING HITLISTS

Our goal in generating hitlists is to share them with other research groups carrying out topology studies. We offer them free-of-charge to all, and to date we have provided them to four other projects. Although hitlists are not human subjects, networks are operated by and involve humans. Hitlist use by multiple prompts us to consider their distribution in the context of the Belmont protocols [27], weighing the benefits and potential costs of sharing and designing policies accordingly.

The benefits of sharing hitlists are similar to sharing of other research results. Shared data is a boon to researchers. A common data source can lower the barrier to entry for future research, and it also makes it easier for researchers to compare their results. (For example, the TREC benchmarks are seen as essential to rapid advances in the field of Information Retrieval [12], although our efforts are far more modest.) As importantly, we expect that the scrutiny of multiple researchers on a common dataset can often identify data or methodological errors that might be otherwise unnoticed. (In Internet topology, the problem of alias resolution is one that is still being refined [2, 20], nearly ten years after the first techniques [16].) For the hitlist creator, a shared result amortizes the operational costs of collection and processing of the input data (Internet censuses) needed to create hitlists. Finally, for the hitlist subjects, network operators in the Internet, a common source allows us to centralize "do-not-probe" blacklists and reduces raw data collection.

Shared hitlists have some costs, however. Most serious is that a hitlist can focus the probing of several researchers on a specific representative address in a network, while independently derived hitlists are more likely to distribute probing load. Second, eventually hitlists will be acquired by malicious users on the Internet. Potential harms are hiding malicious traffic mixed with research traffic, and the slight risk that any list of known active IP addresses may be at risk of additional malicious traffic such as worms or cracking attempts. While a risk, the effort to generate a hitlist is within the reach of a motivated individual, so strong restrictions on hitlists seem unwarranted.

Our current hitlist distribution policies are designed to balance risks with benefits. Although we share hitlists free-of-charge, we provide them subject to a usage agreement. Hitlist users may not redistribute hitlists so we can establish this agreement directly with all users. Tracking hitlist users allows us to estimate load on representatives. We also seed the hitlist with representatives that we monitor to track load. We also hope controlled hitlist distribution delays their acquisition by malicious parties. We expect to review these policies as we gain more experience.

7. CONCLUSIONS

We have defined the properties that are important to hitlists: representatives that are responsive, stable, and provide complete coverage for the Internet. We have developed a fully automated algorithm that mines data from Internet censuses to select informed representatives for the visible Internet. We employ information that is available for about one-third of the Internet, and when an informed representative is available we see it is 50–60% likely to respond 2–3 months later. We showed that the primary reasons for prediction failure are blocks with dynamic addressing and gone-dark blocks that are probable firewalls.

Our hitlists are available free-of-charge and have already been distributed to four different research groups. Although we do not have external evaluation of how their use changes those studies, our evaluation of one prior study suggests the potential to discover 1.7 million additional edge links.

8. REFERENCES

[1] Réka Albert, Hawoong Jeong, and Albert-László

Barabási. Error and attack tolerance in complex networks. *Nature*, 406:378–382, July 27 2000.

[2] Adam Bender, Rob Sherwood, and Neil Spring. Fixing Ally's growing pains with velocity modeling. In *Proceedings of the 8th ACM Internet Measurement Conference*, pages 337–342, Vouliagmeni, Greece, October 2008. ACM.

[3] Randy Bush, James Hiebert, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Testing the reachability of (new) address space. In *Proceedings of the ACM Workshop on Internet Nework Management*, pages 236–241, Kyoto, Japan, August 2007. ACM.

[4] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Internet optometry: assessing the broken glasses in internet reachability. In *Proceedings of the ACM Internet Measurement Conference*, pages 242–253. ACM, November 2009.

[5] Xue Cai and John Heidemann. Understanding address usage in the visible internet. Technical Report ISI-TR-2009-656, USC/Information Sciences Institute, February 2009.

[6] CAIDA. The internet topology data kit—2010-01. `http://www.caida.org/data/active/internet-topology-data-kit/`, January 2010.

[7] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. Internet mapping: from art to science. In *Proceedings of the IEEE Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*, pages 205–211, Alexandria, VA, USA, March 2009. IEEE.

[8] David D. Clark, Craig Partridge, J. Christopher Ramming, and John T. Wroclawski. A knowledge plane for the Internet. In *Proceedings of the ACM SIGCOMM Conference*, pages 3–10, Karlsruhe, Germany, August 2003. ACM.

[9] Eric Cronin, Sugih Jamin, Cheng Jin, Anthony R. Kurc, Danny Raz, and Yuval Shavitt. Constrained mirror placement on the Internet. *IEEE Journal of Selected Areas in Communication*, 20(7):1369–1383, September 2002.

[10] Doug Cutting. Scalable computing with Hadoop. `http://wiki.apache.org/lucene-hadoop-data/attachments/HadoopPresentatio%ns/attachments/yahoo-sds.pdf`, May 2006. Lecture note.

[11] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, pages 137–150, San Francisco, California, USA, December 2004. USENIX.

[12] Alex Dekhtyar and Jane Huffman Hayes. Good benchmarks are hard to find: Toward the benchmark for information retrieval applications in software engineering. In *Proceedings of the 22nd International Conference on Software Maintenance*, Philadelphia, Pennsylvania, USA, September 2006. ACM.

[13] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. Globally distributed content delivery. *IEEE Internet Computing*, 6(5):50–58, September 2002.

[14] Paul Francis, Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. IDMaps: A global internet host distance estimation service. *ACM/IEEE Transactions on Networking*, 9(5):525–540, October 2001.

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

[16] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet map discovery. In *Proceedings of the IEEE Infocom*, pages 1371–1380, Tel Aviv, Israel, March 2000. IEEE.

[17] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and survey of the visible Internet. In *Proceedings of the ACM Internet Measurement Conference*, pages 169–182, Vouliagmeni, Greece, October 2008. ACM.

[18] Bradley Huffaker, Marina Fomenkov, David Moore, and kc claffy. Macroscopic analyses of the infrastructure: measurement and visualization of internet connectivity and performance. `http://www.caida.org/outreach/papers/pam2001/skitter.xml`, November 2001.

[19] Bradley Huffaker, Marina Fomenkov, Daniel J. Plummer, David Moore, and k claffy. Distance metrics in the internet. In *Proceedings of the IEEE International Telecommunications Symposium*. IEEE, 2002.

[20] Ken Keys. IP alias resolution techniques. Technical report, CAIDA, 2008.

[21] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*, pages 367–380, Seattle, WA, USA, November 2006. USENIX.

[22] Harsha V. Madhyastha, Ethan Katz-Bassett, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane Nano: Path prediction for peer-to-peer applications. In *Proceedings of the 6th USENIX Symposium on Network Systems Design and Implementation*, Boston, MA, USA, April 2009. USENIX.

[23] Eric Pfanner. Broadband speeds surge in many countries. *New York Times*, page B8, Oct. 1 2009.

[24] Rob Sherwood, Adam Bender, and Neil Spring. DisCarte: A disjunctive Internet cartographer. In *Proceedings of the ACM SIGCOMM Conference*, pages 303–315, Seatle, Washington, USA, August 2008. ACM.

[25] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with Rocketfuel. In *Proceedings of the ACM SIGCOMM Conference*, pages 133–145, Pittsburgh, Pennsylvania, USA, August 2002. ACM.

[26] Matthew Sullivan and Luis Munoz. Suggested generic DNS naming schemes for large networks and unassigned hosts. Work in progress (Internet draft draft-msullivan-dnsop-generic-naming-schemes-00.txt, April 2006.

[27] The National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. The Belmont report: Ethical principles and guidelines for the protection of human subjects of research. Technical report, Department of Health, Education, and Welfare, April 1979.

[28] USC/LANDER Project. Internet IPv4 address space census. PREDICT ID `USC-LANDER/internet_address_survey_it11w-20060307`. Retrieval information for this and other censuses is at `http://www.isi.edu/ant/traces/`, March 2006.

[29] D.G. Waddington, F. Chang, R. Viswanathan, and B. Yao. Topology discovery for public IPv6 networks. *ACM Computer Communication Review*, 33(3):59–68, July 2003.

[30] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A measurement study on the impact of routing events on end-to-end Internet path performance. In *Proceedings of the ACM SIGCOMM Conference*, Pisa, Italy, August 2006. ACM.

[31] Rich Wolski. Dynamically forecasting network performance using the network weather service. *Journal of Cluster Computing*, 1:119–132, January 1998. Also released as UCSD technical report TR-CS96-494.

[32] Edward Wyatt. Despite ruling, F.C.C. says it will move forward on expanding broadband. *New York Times*, page B3, April 15 2010.

[33] 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*, pages 301–312, Kyoto, Japan, August 2007. ACM.