Do You See Me Now? Sparsity in Passive Observations of Address Liveness (extended)

ISI-TR-2016-710 July 2016

Jelena Mirkovic[1,2] Genevieve Bartlett[1] John Heidemann[1,2] Hao Shi[1,2] Xiyue Deng[1,2]
1: USC/Information Sciences Institute 2: USC/Computer Science Dept.
{mirkovic, bartlett, johnh, shihao, deng}@isi.edu

ABSTRACT

Full allocation of IPv4 addresses has prompted interest in measuring address *liveness*, first with active probing, and recently with the addition of passive observation. While prior work has investigated how to increase coverage by combining multiple sources, this paper explores *what factors affect a passive observer's view*. All passive monitors are *sparse*, seeing only a part of the Internet. We seek to understand how different types of sparsity impact observation quality: the *interests* of external hosts and the hosts within the observed network, the *temporal* limitations on the observation duration, and *coverage* challenges to observe all traffic for a given target or a given vantage point. We study sparsity through *inverted analysis* — a new approach where we use passive observations at three end networks to infer what of these networks would be seen by *virtual* monitors, located at *all* traffic destinations. We show that visibility provided by monitors is heavy-tailed—interest sparsity means popular monitors see a great deal, while 99% see very little. We find that traffic is mostly bipartite, with greater visibility between client-networks and server-networks, than within each group. Finally, we find that popular monitors are robust to temporal and coverage sparsity, but these sparsities greatly reduce power of monitors with initially low visibility.

1. INTRODUCTION

Understanding Internet address use ("*liveness*") is of growing importance, given full IPv4 allocation and growth in address marketplaces. Recent work estimating liveness has observed surprisingly light use of much of the address space [12, 7, 28]. Address liveness also supports studying network topology [10], Internet outages [22], and Internet-level modeling of security phenomena [16, 19, 18].

Prior studies of liveness have used active probing [12, 22]. Recent work has supplemented active probing with passive observation [7, 28] to increase coverage, particularly in regions that do not respond to active probes. Both active probing and passive observation will miss some addresses and blocks. Prior work investigated how various factors (such as probe method and type, duration and filtering) affect the completeness for passive and active at campuses [1] and in the Internet [12, 7, 28]. While that work has established that contributions of multiple passive and active sources are necessary to achieve good coverage, there has been little understanding of what *specific causes* make sources more or less effective, and which portions of the IPv4 space are more or less observable by different sources. Our focus in this paper is to understand what factors affect completeness of *passive observations*. We believe our findings can help researchers improve collection strategies, interpret observations, and clarify sources of imprecision inherent in measurements of the full Internet.

We assume a *monitor*, placed at some vantage point, assesses liveness for a given *target* network. The monitor's passive observation of the target through the traffic it sends provides some *visibility* into the target's live addresses.

The first contribution of our paper is to identify *sparsity*, the properties of the monitor and target that limit visibility. Sparsity takes several forms. *Interest sparsity* reflects how much users care about content. An observer at a popular website will see many more addresses, than an observer at a rarely visited site. *Temporal sparsity* follows from the finite duration of any observation, which may miss infrequently used addresses. *Coverage sparsity* occurs when traffic for a target evades observation, for example if only some links for an organization support monitors, due to multi-homing or use of different media (wired and wireless). A variation is *sampling sparsity*, where observations are down-sampled to handle high line rates.

Our second contribution is to develop a new measurement methodology, *inverted analysis*, to understand sparsity at Internet scales. We use traces of outgoing traffic from three U.S. universities as ground truth. We treat these networks as our measurement targets, and place "virtual" observers at *all other network prefixes*. These virtual observers tell us what a passive observer

there would see of our targets. Inverted analysis is essential to generalize from our targets to what one could see around the Internet.

Our combination of many virtual observers and traffic analysis helps us understand *why* some observers see more or less of our target networks. As the final contribution, we demonstrate that *interest sparsity* has the strongest effect on completeness of passive observation. While other aspects of sparsity can be controlled (observe longer, capture all links, and avoid sampling), interest sparsity reflects the inherent nature of a vantage point's location and traffic patterns. We show that human interest strongly affects passive observations, and is essential for observation completeness. A few networks, which host highly-popular content have much higher visibility into our targets, than the rest of the Internet – leading to heavy tail. We further show that networks, which prevalently host clients form nearly bipartite communications patterns with those networks which consist mostly of servers. This means that visibility between client-heavy networks and server-heavy networks is larger, than the visibility within each network category.

2. PROBLEM STATEMENT: LIVENESS AND SPARSITY

We next frame the problem we study: we define liveness, discuss how it can be measured, and discuss sparsity's effects on measurement.

2.1 Evaluating Liveness

The definition of liveness depends on the source of liveness information and the purpose of estimation. *Cumulative* liveness denotes a target as live if it is active in any available data source, within some long time interval. Such liveness is easily estimated from passive observations [8, 28] or from multiple Internet censuses, or a survey with repeated observation [12], and is useful for studies of the Internet's address space utilization. *Instantaneous* liveness represents what is live in a snapshot at one time. It can be collected over months (a virtual snapshot [5], as in censuses), but each address gets one "try". Liveness can be assessed via *counts* of live addresses, or one may want to learn the *exact identities* of the computers using those addresses, or even the services they run. One may also study liveness of *blocks* of adjacent addresses, typically defined by the length of the IPv4 address prefix they have in common

This paper examines *cumulative counts of liveness of addresses and /24-prefix blocks* (aka "blocks" for short) using passive observations. Our approaches also apply to IPv6. Because we have low volumes of IPv6 traffic in our datasets, we leave its study for future work.

2.2 Passive and Active Measurement Methods

Researchers have studied Internet liveness through

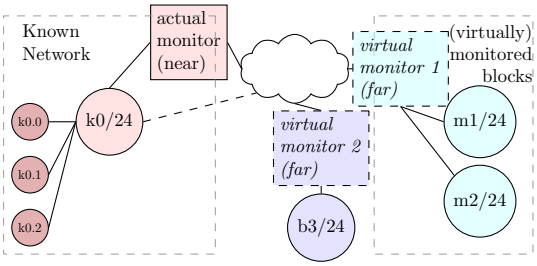


Figure 1: A monitor next to a Known Network (left) and two virtual monitors (right and bottom).

passive observations, active probing [13, 9], and their combination [1, 7, 28].

Active measurement sends probes (often ICMP echo requests or TCP SYNs) to some or all addresses, and recognizes those that reply as live. Internet censuses cover the entire IP space, typically in one pass, while surveys cover a fraction, but via repeated, more frequent probes. One-pass active probing captures an instantaneous count of live addresses that are willing to respond to probes, while repeated censuses or surveys can build cumulative counts.

Passive observations record traffic that passes a *monitor*. That monitor records packets or flows (pcap, netflow, and Argus are common formats); the source addresses of traffic that transits the monitor are noted as live. Passive monitors are vulnerable to spoofing (forged source addresses).

A passive monitor may reside at a border router of an end-network and observe all traffic between this network and the outside world [11]. In Figure 1, the left region labeled Known Network is such a monitored network, with a monitor between it and the Internet. Multiple peerings (such as the dashed link from k0 to the Internet) result in incomplete observations (§ 6.3).

We obtain traffic from several *real* monitors in front of end networks, as described in § 3.1. In § 3.2 we develop inverted analysis, an approach that allows us to study what can be seen of our Known Networks, when observations are made near their traffic's destinations. We therefore talk about *virtual monitors* in those far networks, and what they observe of our Known Networks as *targets*. Virtual monitors are boxes with dashed lines on the right of Figure 1. Monitor 1 sees traffic sent from k0/24 to blocks m0/24 and m1/24, and monitor 2 sees traffic sent from k0/24 to b3/24. When we highlight the location of a monitor, we use the term *near monitor* for data we collect at our Known Networks, and *far monitor* for virtual monitors placed around the Internet.

We can generalize monitors in two ways. First, although monitors are at the Internet gateways of edge networks in our datasets, in principle they could be placed at backbone networks [4, 3]. With asymmetric routing, monitors on backbones will have a high degree of coverage sparsity. Second, in addition to directly ob-

served packet data, one could use any kind of logs [24, 25] to infer liveness of sources, which created log entries. In this case, the "monitor" is effectively the aggregate of all end systems providing logs.

We define the *visibility* $V_{m,t}$ of a monitor m with respect to a given target t as the percentage of t's live addresses or /24 blocks that are observed by m. In other words, $V_{m,t}$ is the fraction of ground truth m is able to learn about t. We define *sparsity* as the limitations of the monitor that reduce visibility: $S_{m,t} = 1 - V_{m,t}$.

Different monitors will see different fractions of a target. We find it helpful to group monitors by their "power": the fraction of a given target, which they observe. We group monitors into regions of *low*, *medium*, *high*, and *near-complete* visibility defined as less than 1%, 1–10%, 10–50%, or more 50% of the active addresses or /24 blocks for a given target. These terms are defined relative to the actual number of target's addresses, which were active over the measurement period (not allocated), and is defined (from a theoretical point of view) even if that number is not known. In practice, we evaluate it using the best-available ground truth for our targets, provided by our real monitors.

2.3 Sparsity: the Challenge for Passive

We seek to quantify how different types of sparsity impact the visibility of targets and if some targets or monitors are impacted more than others. We identify four types of sparsity:

Interest sparsity: a monitor at the far network does not observe the target, because end clients from the far network are not interested in the content served by the target network, and vice versa.

Temporal sparsity: a monitor does not observe the target because the target is intermittently active and the monitor's observation is not long enough.

Coverage sparsity: a monitor does not observe the target because it does not observe links traversed by some of the target's traffic to the far network.

Sampling sparsity: a monitor does not observe the target because it samples packets or flows to reduce load. The target sends a small number of packets, and the monitor's sampled observations miss all of them.

2.4 Clients, Servers and Promiscuity

To understand visibility and causes for interest sparsity, we must examine how a target's addresses and blocks send traffic to external networks. This includes traffic sent to initiate communication, and responses to requests sent by external networks. Traffic comes from *clients* and goes to *servers*. These are properties of traffic, but we find many addresses act mostly as a client or mostly as a server, some perform as both (§ 5.1), and a few are neither (e.g., they only send ICMP probes). We describe how we classify addresses in § 3.3.

The contribution of each address to visibility depends on its *promiscuity*: how many other addresses it talks to. Promiscuity indicates traffic exchange, and is a property of both clients and servers. Clients' promiscuity is driven by the interests of their users who initiate communications. Servers respond to traffic, with their promiscuity driven by the popularity of their content. A scanner walking the IP address space or a Web crawler are examples of promiscuous clients, since they talk to many external networks. The Alexa Top-100 websites are hosted on promiscuous servers; they are visited by many clients from external networks, and in turn send responses to them, which makes them visible in passive traces.

This paper looks at edge networks, but others have considered passive observation at backbones. Backbone monitors on routes leading to promiscuous clients and servers will have good visibility of many targets.

3. METHODOLOGY

We next describe how we study liveness: our data sources, how we use virtual monitors around the Internet, and how we classify traffic and addresses.

3.1 Data Sources

We use five passive datasets in our study, from four sources: three week-long traces capturing much of the traffic at the edge of three US universities, and two traces from a major U.S-based CDN. We call these subjects our *Known Networks*, and use them as both targets and monitors at different times in our work. Table 1 gives times and durations of each dataset, the size of the Known Network, and how many external blocks it sees.

Each trace has all addresses anonymized, with the 24 most-significant bits unchanged and the remaining 8 bits cryptographically scrambled [27]. (We omit IPv6 from analysis at this time, see § 2.1). Anonymization is consistent for each dataset but not across sources, allowing for comparison of /24 blocks but not specific addresses, across datasets.

The three universities in Known Networks are all of similar sizes: each about 30,000 students. *U.Ga.* and *CSU* operate /16 networks, and *USC* operates two /16s. All collect Argus-format flow data [17].

We evaluate completeness of data collection (coverage sparsity) in § 6.3. We show that *USC* is near-complete, but at *CSU* we consistently miss some blocks . For *U.Ga.*, we also see no external ICMP traffic, and they also employ NAT for many of their clients (around 95 K addresses from 172.16.0.0/12). Our collector is inside the NAT and we do not know the mapping of private to public IP addresses, so we discard all NAT'ed traffic in both directions.

Our CDN data uses logs from webservers, summarized to only show client /24 blocks and anonymized client addresses. We have two datasets; each sampled differ-

Known Network		Observation					Known Net Size		External
organization	prefixes	Format	Start	End	Duration	Flows	IPs	blocks	blocks
USC	128.125.0.0/16, 68.181.0.0/16	Argus	2014-06-17	2014-06-23	7 days	461 M	31,997	492	6.2 M
UGA	128.192.0.0/16	Argus	2016-02-06	2016-02-18	13 days	682 M	5,243	198	0.7 M
CSU	129.82.0.0/16	Argus	2014-06-17	2014-06-23	7 days	2.2 B	17,732	186	4.4 M
CDNXpop	—	logs	2014-06-17	2014-06-23	7 days	266 B	—	—	—
CDNXglobal	—	logs	2014-06-17	2014-06-23	7 days	200 B	—	—	—

Table 1: Datasets used in this paper.

ently due to high traffic volumes. *CDNXpop* takes all log entries from all servers at points-of-presence in Los Angeles and Chicago, then down-samples to save roughly 1 record in 1,000. The data is recorded continuously over one week. *CDNXglobal*, instead covers all PoPs (more than 30, at the time), and records all queries, but only for 1 hour per day, every day for a week. The specific hour is chosen to match peak traffic period, at each PoP's location. Sampling makes analysis of our CDN dataset difficult, but we compare the traces and scale to approximate unsampled data (§ 5.3.1).

Limitations of these sources: Our data sources have certain limitations. Each covers a tiny part of the entire Internet and so may not be representative. However, our universities have both clients and servers, while *CDNX* is a large real-world CDN. Our datasets thus have diversity of clients and servers, both as targets and as observers.

Our observations are only one week long, while prior work includes data for months or years [28, 7]. Longer data is essential to provide estimates of the number of live addresses in the Internet, but our goal is instead to understand the *reasons* for where passive observations works well or poorly. These reasons are not strongly dependent on observation duration.

Our observations are sparse in several ways: we do not get complete traffic for all Known Networks (although we believe we get most of it), and we have some measurement loss for *USC* during peak load (the busiest hour of the day). We evaluate the effects of these kinds of sparsity in § 6.3.

While more data is always helpful, we believe our five datasets from four different sources are sufficient to support our findings about sparsity and its effects on visibility of passive monitors.

3.2 Inverted Analysis: Who and How Much

We have access to monitors for only four networks (Table 1), and by themselves they cannot observe much of the Internet. However, *inverted analysis* enables us to place *virtual* monitors at all networks to which our Known Networks send traffic. This enables us to study *what the world sees of our Known Networks.*

Consider the two dashed virtual monitors in Figure 1. We have a real monitor covering what the Known Network k0/24 sees of the world. Using observations of outgoing traffic at this monitor, we infer what far virtual monitors m1/24 and m2/24 see of k0/24.

3.3 Classifying Flows, Address, and Blocks

We have several goals in analyzing our flow and log data. First, we use it to identify live addresses, and from those live blocks at our Known Networks. But to understand the reasons why we see or miss addresses, we also classify traffic flows by their purpose (e.g., scan, ICMP probe, etc.) and the role a Known Networks' address plays in the flow (e.g., server, client). From this classification, we can infer the dominant traffic for each address in the Known Network (client, server, etc.), and from there, classify addresses and blocks by their common usage.

3.3.1 *Cleaning Flow and Log Data*

We first must identify and correct for sources of error in our data formats. For CDN log data (in *CDNX*), a request record indicates an established TCP connection, so we immediately declare the source to be live.

Argus flow data requires much more care than log data, because its summarization of a flow discards information, particularly in the face of packet loss or reordering at the monitor. Argus flow records indicate its understanding of a TCP connection's source and destination, and its dynamics, but discard packet-level details (for example, we lose information needed to differentiate between a transmission of two SYN-ACK packets versus, a SYN followed by an ACK).

We found several situations where Argus would switch the source and destination addresses in a flow record. When Argus processes out-of-order TCP and UDP packets (even if their timestamps indicate the misordering) it flips the source and destination in the resulting flow record, along with other features such as ports and TTL. For ICMP ECHO Requests, out-of-order packets also cause a source/destination flip, but the type and code fields are not flipped. In some cases we can identify and correct Argus' mislabeling. Overall, 13.9% *USC*, 2.2% *U.Ga.* and 21.3% *CSU* TCP and UDP flows were flipped. We use IANA-defined service ports to identify clients and servers in TCP and UDP flows and correct Argus' source and destination assessment. We apply this correction to all TCP and UDP flows, which we have identified as flipped. We observed similar levels of ICMP mislabeling (by measuring mislabeling on flows from known active probers), but unlike the TCP and UDP cases, we have no way to identify and correct mislabeled flows. We therefore avoid using analysis which relies

on differentiating between the source and destination of ICMP flows.

3.3.2 From Data to Flow Types and Liveness

We next classify flows to identify their purpose ("flow's purpose" in Table 2) and the role of our target's address in the flow ("target address's role" in Table 2).

In passive observations, some flows may be spoofed. Spoofing does not affect our inverted analysis as we use traces collected near our targets, and observe only flows from our assigned address ranges. While a scanner within our target could subnet-spoof a different address, we do not see many outgoing scans and thus believe this type of spoofing does not affect our findings. In § 5.3 we consider what we see of the world, and there our observations may include spoofed traffic. There we apply statistical analysis, as outlined in [28] to correct our observations for spoofed traffic.

We infer *clients* and *servers* based on well-known service ports [14]. We mark flows as *ambiguous* when both ports are service ports, or both are client ports. Such flows may be peer-to-peer traffic or communication between NTP or DNS servers. In principle, we could identify client and server from the flow direction, but packet reordering and Argus flipping of the source and destination labels provides noise, which interferes with this approach.

3.3.3 From Flows to Address Types

Going from flows to addresses, we label each address by aggregating address-role labels of its flows. We preferentially choose labels that carry more information about an address: choosing client or server over other role labels. Addresses, which participate both in client and server flows are labeled as *client-server*. Those that only have ambiguous, echo-src, responder or unidirectional flows are labeled as *ambiguous*, *prober*, *responder* and *oneway source*, respectively.

3.4 Statistical Corrections for Spoofing

For analysis in § 5.3.1, we can account for spoofing statistically by adapting the technique from Zander et al. [28] to our datasets. We identify known unused address space (about 50 /8 blocks), then look for traffic from those addresses to determine the mean spoofing rate s. We find about 17 K spoofed addresses per unused /8 in the entire *USC* dataset, but only 900 in *U.Ga.* and 50–150 in *CSU*. This difference suggests that *U.Ga.* and *CSU* actively filter unsolicited traffic. If we assume spoofing uses all IPv4 addresses with uniform probability, we can then compute the probability that any packet is spoofed as $p = s/2^24$. We find that p ranges from $5 * 10^{-8}$ to 0.001 for our datasets.

Since we study counts of live hosts we do not need to identify and remove the exact spoofed addresses. Instead we correct our live address/block count estimates for spoofing by discarding each live address as statistically spoofed with probability p, and discarding blocks where all addresses are statically spoofed.

4. THE ROLE OF INTEREST SPARSITY

We now show that interest sparsity significantly impacts visibility. When vantage points (VPs) are placed at edge networks, VPs placed at the most popular networks will see far more than VPs placed at randomly chosen networks.

We then investigate the effects of aggregating observations from multiple monitors, and show that gains are limited, unless the monitors initially have high visibility.

These two results point to *interest sparsity*: only "interesting" content results in powerful monitors. We explore a possible cause for interest sparsity in § 5.

4.1 Visibility is Heavy Tailed

The more popular the content at a monitor, the more targets are attracted to the monitor, increasing the visibility the monitor has. We show here that the "popularity" of a monitor's content is the dominant factor in what it sees—*this visibility is heavy tailed.*

We evaluate the power of different monitors with inverted analysis of our three University targets, asking how much of each target would be seen by all possible external monitors. We place a virtual monitor at all possible /24 blocks that exchange traffic with each of our three targets, and evaluate how many of the target's addresses and blocks each monitor would see. Figure 2 shows the the log-log complementary CDF.

These graphs show only the visibility of blocks that see our targets. Most of the Internet does not see them at all: there is no interaction between 93% of routable blocks and *U.Ga.*, 58% and *CSU*, and 41% and *USC*. These calculations are based on the size of the routed address space at these observation periods, as reported by RouteViews [26] (10,537,665 in 2014, and 11,000,925 in 2016).

Both graphs show an inflection point, where a handful of monitors see half of each target, then a basically linear region where visibility falls off as a heavy tail over about three orders of magnitude. The tail is weakest (shortest and most inflected) for University of Georgia. *U.Ga.* has far fewer hosts in its globally routable prefix (the rest of *U.Ga.*'s addresses are private addresses, which are NATted and thus we do not include them in our analysis) than the other targets (see Table 1), and those hosts are seen by far fewer external monitors (0.7 M, compared to *CSU*'s 4.4 M and *USC*'s 6.2 M), which leads to the tail weakness.

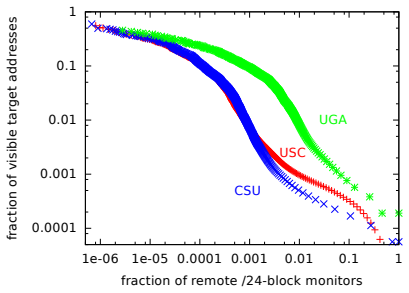
Conclusion: *The visibility of monitors is heavy tailed.*

4.2 Whole ASes Improve Visibility

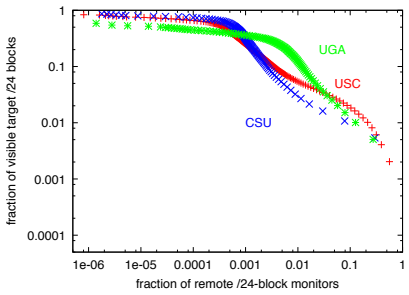
When we look at monitors observing traffic from a

Flow label	Proto.	Criteria
Flow's purpose		
useful	T/U	TCP PSH in both directions or two-way UDP flow
scan	T/U	Pkt to a service port. For TCP it must be a TCP SYN pkt
TCP scan response	T	Pkt from a service port, received TCP SYN, replied with TCP SYN ACK or TCP RST
backscatter response	T	Pkt to a service port, received TCP SYN ACK, replied with TCP RST
partial	T/U	all other T/U flows
echo/unreach	I	ICMP flows
Target address's role		
client	T/U	send pkts to a service port (which receive some reply)
server	T/U	receive pkts on and reply from a service port
ambiguous	T/U	both ports on the flow are a service port, or both are a non-service port
echo-src	I	send ECHO pkts and receive replies
echo-responder	I	receive ECHO pkts and reply to them
rst-responder	T	receive pkts on a service port and reply with TCP RST with optional ACK bit set
unidirectional	T/U/I	no reply from Far IP
Final flow classification		
human-interest	T/U	(client or server) and useful
unsolicited	T/U/I	((client or server) and (scan or scan response or backscatter response)) or echo-src or echo-responder or rst-responder or oneway
undetermined	T/U/I	ambiguous or other

Table 2: Labeling flows by their behavior. Protocols are TCP, UDP, or ICMP.



(a) Addresses



(b) /24 blocks

Figure 2: Visibility of addresses and blocks from all possible virtual monitors, placed in networks that exchange traffic with our targets.

/24 block of addresses, visibility is heavy-tailed. Many organizations run networks that are larger than 256 IP addresses, so we next consider what happens if we place monitors that cover an entire AS. We evaluate this as a thought experiment since ASes represent administratively distinct entities some of which may be large, geographically distributed and difficult to monitor in their entirety.

We enumerate all ASes that see our targets by starting from each /24 block that receives traffic from a target, identifying that block's AS from Neustar Web API [20], then identifying all other prefixes that belong to that AS through IPInfo [15]. Figure 3(c) shows the distribution of sizes of these ASes, as a count of /24 blocks.

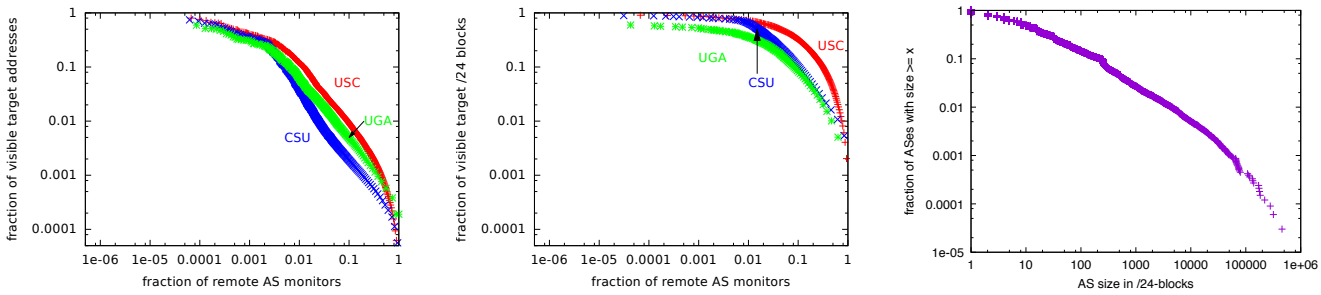
Figure 3 shows CCDF for how much of our targets is seen from AS-sized monitors with graph scale held the same as Figure 2. We see that visibility of both addresses and blocks improves, in that the curves shift up and left. Between 2.42% and 9.3% of organization-level monitors see more than 1% of our target's IPs (compared to 0.08-0.6% of block-level monitors), and 39–69% of organizations see more than 1% of our target's /24 prefixes (compared to 26-29% of block-level monitors).

4.3 Promiscuity Enables Visibility

To understand *why* visibility is heavy tailed, we consider communication dynamics of the target and the far network. We find that *promiscuity* of targets and of hosts in the far network is what affects visibility.

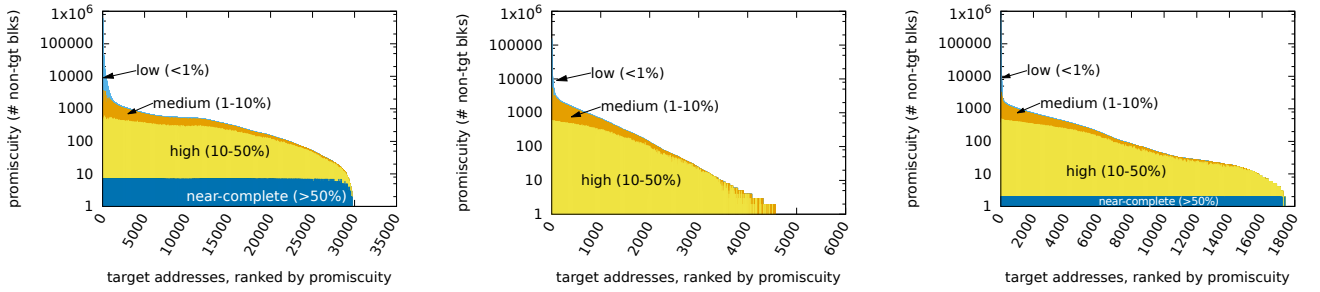
Promiscuity of target addresses: Target addresses that talk to many external sites are more widely seen. Figure 4 ranks each address in a target by how many external /24 blocks it exchanges traffic with. That many block-level monitors will observe the given address. On a log-linear graph, we see that a few target addresses speak with *many* external blocks. The first 6–14% addresses in each target interact with 1,000 or more external /24 blocks (10% for *USC*, 14% for *U.Ga.*, and 6% for *CSU*). We expect that these are popular Web servers contacted by many external clients, Web proxies for the university, or scanners. Because of their high promiscuity, they will be seen by many far monitors. The majority of addresses in each target talk to a small number of external networks in a week, and thus will not be widely seen by far monitors. These addresses are likely client machines and servers serving less popular content.

Promiscuity of blocks in the far network: Promiscuity also applies to addresses and blocks on the "other side" of the monitor, in the far network. The stacked



(a) Visibility of target addresses (b) Visibility of target /24-blocks (c) Distribution of AS sizes

Figure 3: Visibility of addresses and blocks from all possible remote ASes as monitors. (For ASes that receive traffic from our targets).



(a) Target: *USC* (b) Target: *U.Ga.* (c) Target: *CSU*

Figure 4: Promiscuity of addresses in each target to all far monitors, shown sorted by descending rank. Colored layers indicate the visibility power of far monitors by category (low, medium, high, or near-complete).

layers in Figure 4 each show the visibility range (low, medium, high or near-complete) of each far monitor with regard to the given target. We see that the most powerful remote networks (the blue band at the base of the stack) see almost all addresses—these networks run scanners, popular web services, or update services for common software (§ 5.2.1).

While many IPs talk to a few common /24 blocks, the rest of communications is scattered among a large number of blocks, with each having high to moderate visibility into our targets. A complement to these powerful remote monitors, are low-visibility monitors, which show as a thin light blue band at top of the stack. These monitors see only a few target addresses, and those that are seen are usually very promiscuous.

Figure 4 shows promiscuity of target blocks. We see similar results as for addresses, and omit their discussion due to space.

Conclusion: *Promiscuity in target and monitor addresses and blocks is* the *dominant factor affecting visibility.*

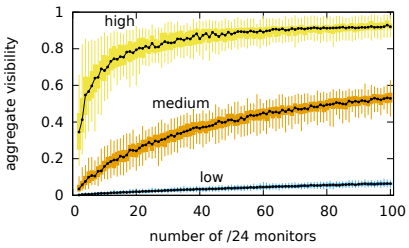
4.4 More Monitors Do Not Help

We have shown that promiscuity determines the heavy-tailed nature of visibility—a few addresses in the target are visible everywhere, and a few remote monitors see a great deal. The implication of this result is that *more monitors show limited benefits*—what matters is *monitors which observe in promiscuous networks.*

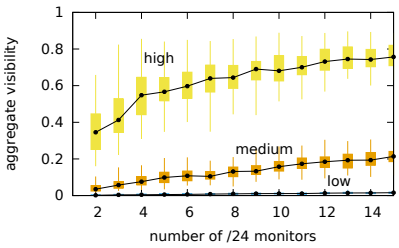
To quantify the effect of more monitors, we pick a random set of n monitors, all uniformly chosen from monitors with the same level of visibility (low, medium, or high). Figure 5 shows how the addition of monitors improves visibility. Each point in this graph represents 100 iterations of the experiment, with the lines showing the median values, boxes quartiles, and whiskers minimum and maximum. Complete visibility (1 on graphs) here is the estimated number of live addresses in the target based on observations from our near monitors. (We report data for *USC* as a target; other targets show similar trends.)

We first observe that there is a *huge* advantage to selecting monitors with stronger visibility. The visibility power of any given monitor depends strongly on the popularity of its server content and the activity of its clients, i.e. promiscuity of far networks dominates visibility.

Second, we see that adding a few monitors shows considerable benefit in each class. Adding a second, third, or forth monitor can improve visibility by 2× to 4×. While the effect is far smaller than replacing a given monitor with a more popular one, diversity from a few sites of the same popularity is helpful.

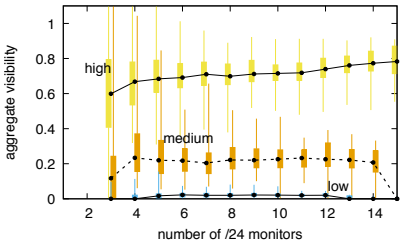


(a) Adding from 1 to 100 /24-block monitors

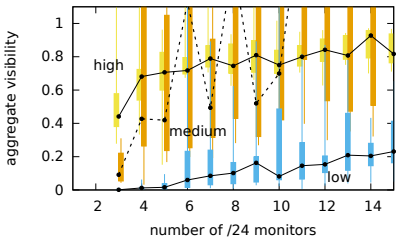


(b) Aggregated, 3–15 sources

Figure 5: Visibility as numbers of block-monitors grows. Monitors are drawn from different classes of visibility (low, medium or high). Lines show median, boxes quartiles, and whiskers minimum and maximum. Target: *USC*.



(a) Best of two Poisson models



(b) Best of all models in Rcapture package

Figure 6: Visibilty with capture-recapture estimation, as numbers of /24-block monitors grows. Monitors are drawn from different classes of visibility (low, medium or high). Lines show median, boxes quartiles, and whiskers minimum and maximum. Target: *USC*.

Finally, we see that additional coverage quickly reaches diminishing returns. There seems relatively little benefit with more than 20 high-visibility monitors. With medium- or low-visibility monitors, there is much more room to improve, but improvements still diminish as more monitors are added.

Conclusion: monitor *power*, not *quantity* matters. *Only twenty high-visibility monitors are required to see most of our target networks* (80%), and adding more does little to increase coverage.

4.5 Better Passive Analysis: Capture-Recapture

More monitors show diminishing improvements in visibility, but can we make *smarter* use of the monitors we have? Capture-recapture analysis uses estimates about the incremental discovery by each new monitor to predict the total population size, with promising results at improving estimates of network visibility [28].

To apply capture-recapture analysis, we select n random block-monitors from different visibility ranges (as in § 4.4), but we use capture-recapture technique to estimate total number of addresses in our targets. Choice of statistical model is important in capture-recapture, and standard practice is to select the model with the smallest BIC (Bayesian Information Criterion), breaking ties in BIC by selecting the model with the smallest degree of freedom. We apply this approach two ways, first considering two Poisson models, as in prior work [28], and then considering all available models in the R software's Rcapture package [23]. We take 100 random trials for each experiment, but are forced to use only 20 trials with 12–15 monitors due to high computation cost. In each case we plot median, quartiles, and minimum and maximums in Figure 6.

Poisson models: We see that with Poisson models (Figure 6(a)), capture-recapture provides considerable benefit for a few monitors—estimates with two or three monitors approach the actual visibility of about six monitors using only direct observation (compare Figure 6(a) with Figure 5(b) for any type of monitor). Although capture-recapture converges more quickly than brute force, it seems to converge on about the same limit.

Additional models: With additional models, capture-recapture results in much higher estimates than mere aggregation of monitor observations. The gain is especially high for low and medium-visibility monitors.

However, using more than just Poisson models adds a great deal of uncertainty to the results, as manifested in the large quartiles (the yellow, orange, and blue boxes) in Figure 6. This is especially notable for medium-visibility monitors, which have huge quartiles and change the median value. In addition, many estimates drawn from medium-visibility monitors give results that are much larger than our estimated ground truth. Samples from high- and low-visibility monitors show a more consistent median, but still have wide quartiles.

We caution that capture-recapture analysis poses a risk. Its analysis depends on assumptions about the independence of observers and the target. It is not clear that that independence is warranted, as shown by the

high degree of noise we see in capture-recapture results. The root cause of this noise is likely large variation in address promiscuity (§ 4.3) and the heavy-tailed visibility of monitors (§ 4.1), both of which violate the assumption of independence. This problem is at its worst for medium-visibility monitors, where the amount of overlap between two monitors can vary a great deal, leading to very different estimates. It is possible that capture-recapture can be applied by carefully choosing monitor locations to improve independence, or by carefully validating the results against known targets, but both seem challenging to estimate.

Conclusion: *Capture-recapture increases estimates of visibility with few monitors*, but it provides *limited benefits as the number of monitors increase*, and it *risks introducing large uncertainty* into estimates.

5. BIPARTITE TRAFFIC AND INTEREST SPARSITY

We showed that monitor visibility is heavy tailed, and the biggest factor that affects visibility is promiscuity: addresses that interact with many others. But *why* do some address interact with many others? And *which* others do they they interact with?

We next show that most addresses are either *clients* or *servers*, but rarely both, based on their wide-area traffic. This leads to mostly *bipartite* traffic patterns, with more communication, and greater visibility between client-heavy (eyeball) networks and server-heavy (cloud) networks, than within each network category. This bipartite structure has a strong influence on visibility, because while there are promiscuous addresses that are either clients or servers, neither has good visibility into others of its own type.

We perform our analysis by first classifying addresses in our targets (§ 5.1). This helps us understand the nature of far monitors, and how much of each address class they see (§ 5.2), and establish this bipartite structure.

We then reverse this approach, asking what our datasets, consisting of monitors at three client-heavy networks (*USC*, *U.Ga.* and *CSU*) and one server-heavy network (*CDNX*) see of the world (§ 5.3). This confirms the bipartite structure.

5.1 Classifying Addresses in Our Known Networks

Table 3 shows our classification of addresses in our targets, using rules from Table 2. We see that our networks include many clients: with 55–71% of addresses sending client traffic only. However, we see a smaller but significant number of servers (4–14%), and a number of addresses that send a mix of client and server-like traffic (10–22%). Finally, we see very few addresses which send no client or server traffic, but send ambiguous flows, or send/respond to unsolicited traffic. The mix of clients and servers in university environments makes our targets ideal for our study.

address class	**USC** addrs.	*blocks*	**UGA** addrs.	*blocks*	**CSU** addrs.	*blocks*
CO: client-only	55%	*86%*	52%	*62%*	71%	*87%*
SO: server-only	12%	*63%*	14%	*68%*	4.4%	*39%*
CS: client-server	21%	*77%*	22%	*51%*	10.1%	*72%*
A: ambiguous	1%	*26%*	2%	*21%*	0.7%	*32%*
P: prober	3%	*48%*	0%	*0%*	0.2%	*2%*
R: responder	1%	*12%*	4%	*21%*	13.3%	*83%*
OW: oneway src.	6%	*59%*	5%	*35%*	0.3%	*22%*

Table 3: Address and /24-block classification at three targets: *USC*, *U.Ga.* and *CSU*.

Looking at the compositions of /24 blocks, addresses that are clients and servers are spread over all blocks. This diversity in location means most blocks include both clients, servers and addresses sending mixed client-server traffic. Only *CSU* shows larger homogeneity across blocks, with servers and clients being more segregated. This diversity in each block makes it likely that all classes of remote monitor will see many prefixes, even if they touch only certain addresses in each prefix.

5.2 Who Sees Classes in Our Targets

Having classified addresses in our targets, we next look at far monitors to find what they see. We start by looking at those that see the most of our targets, *heavy see-ers*. We then turn to a random sample of far monitors for an unbiased characterization.

5.2.1 The Heavy See-ers

We begin by looking at far monitors that have the greatest visibility into our targets. Such monitors are the most important in any study using passive observation.

Table 4 shows the ten top, block-monitors and what they see of our targets, ranked by their visibility of target addresses. Monitors are identified by their AS names, and duplicates are common for providers who have many address blocks. We further report the top flow label in the last column of Table 4.

We see that the top possible monitor is always an academic network close to our targets. Using reverse DNS for these blocks, we confirmed that they host caches for Google, Netflix and Akamai. Going further down the list, the remaining top monitors are mostly large content providers and CDNs: Google, Akamai, Facebook, and Edgecast.

Finally, we see that these monitors see clients and mixed client-servers in each target, due to client traffic sent by our targets. The exceptions are WIDE for *USC* and Secured Servers for *CSU*, which also see servers and responders. The broad coverage of WIDE is due to ICMP ECHO probing, and that of Secured Servers is due to their scanning of port 443. The large visibility of oneway sources in *U.Ga.* by all monitors occurs because of asymmetric TCP traffic, which we classify as unidirectional.

Top Far Monitors	% seen addrs	blks	% address classes seen CO	SO	CS	P	R	OW	why
target: *USC*									
CSU Net	63	79	90	0	66	0	0	0	cli-usef
Google	56	77	79	0	58	0	0	0	cli-usef
Google	56	77	79	0	58	0	0	0	cli-usef
Akamai	55	78	77	0	60	0	0	0	cli-usef
NTT	51	78	72	0	54	0	0	0	cli-usef
Edgecast	50	77	71	0	53	0	0	0	cli-usef
Akamai	50	77	70	0	55	0	0	0	cli-usef
Edgecast	47	77	68	0	48	0	0	0	cli-usef
Facebook	47	77	67	0	48	0	0	0	cli-usef
WIDE	46	75	34	85	63	89	16	0	echo-resp
target: *U.Ga.*									
Georgia Tech	92	74	95	0	89	0	0	46	cli-usef
Google	92	72	94	0	88	0	0	40	cli-usef
Google	89	72	92	0	86	0	0	37	cli-usef
Google	89	71	91	0	86	0	0	37	cli-usef
Google	88	71	91	0	86	0	0	36	cli-usef
Facebook	88	69	90	0	84	0	0	32	cli-usef
Google	85	71	87	0	84	0	0	34	cli-usef
Google	83	70	85	0	82	0	0	32	cli-usef
Edgecast	77	70	79	0	78	0	0	34	cli-usef
target: *CSU*									
UCAR	69	83	87	1	71	0	0	18	cli-usef
Google	59	82	74	1	61	0	0	1	cli-usef
Google	49	81	63	0	43	0	0	0	cli-usef
Secured Svrs.	49	87	40	34	59	14	95	0	rst-resp
Time Warner	48	80	62	0	43	0	0	0	cli-usef
Edgecast	47	80	61	0	36	0	0	0	cli-usef
Edgecast	46	80	59	0	42	0	0	0	cli-usef
Google	43	81	57	0	33	0	0	0	cli-usef
Facebook	43	80	55	0	34	0	0	0	cli-usef

Table 4: Ten block-monitors with the largest visibility of addresses. Their visibility into blocks and addresses-by-class is also shown, as well as top cause of visibility. Far monitors are identified by their ASes.

Conclusion: The analysis of our three targets show that server-heavy monitors have excellent visibility into our client-heavy networks.

5.2.2 *What Most Far Monitors See*

We saw that some server networks have high visibility into our client-heavy targets (due to high promiscuity, § 4.3), but that *in general* many far monitors have low visibility (§ 4.4).

To understand the relationship between visibility and a nature of far network (server-heavy or client-heavy) we need a set of /24-blocks that provide a range of types of networks with different levels of visibility for our university targets. Thus, for each university, we randomly select 300 /24-blocks that see that it, 100 blocks from each of the low, intermediate and high visibility range. We call this set the *Random Far Monitors*.

We then manually classify each Random Far Monitor as client- or server-heavy. We first identify its AS and organization, and then examine information from the organization's web pages, the company overview page of the Bloomberg site (if it exists), [2] and the PeeringDB [21]. We classify ASes owned by hosting networks (Secure Servers, Fastly, etc.), CDNs and content providers (Akamai, Dropbox, Facebook, etc.), and enterprises (such as banks and news sites) as server-heavy. ASes owned by organizations, which provide connectivity (T-Mobile, Comcast, India Telecomm) or host

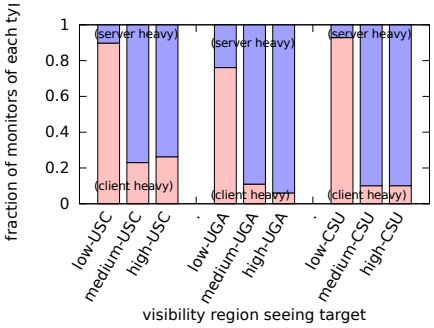


Figure 7: Visibility into each target (the three groups), from far /24-block monitors of different visibility classes (left, center and right bars of each group), with far monitors identified as client- or server-heavy (the top and bottom of each bar).

large numbers of users (such as universities) we label as client-heavy. When classification is unclear, we follow the primary service presented on the organization's web page.

Figure 7 shows the visibility into each target by the Random Far Monitors, with groups of three bars by target, and in each group, bars broken out by the visibility-region of the monitor. All three targets are seen in similar ways. Remote monitors at client-heavy networks tend to have poor visibility into each target (the left bar of each group is mostly client-heavy monitors). To get good visibility into these targets, one needs to operate a monitor at a server-heavy network (the right two bars). This analysis generalizes the observations from our last section, that mostly servers have good visibility of our client-heavy targets. It is the first step in showing that network traffic is largely bipartite.

To confirm our assessment of causes of visibility, we look at the top flow label for traffic going from our targets to each of our Random Far Monitors. For space reasons, we only discuss what causes *USC* visibility by Far Random Monitors. In the high-visibility region, 84% of server-heavy monitors see us due to our client traffic. The remaining 16% scan us, and receive TCP resets, ICMP ECHO replies or SYN ACKs from us. Further, 38% of client-heavy monitors in this region see us due to our client traffic. These networks are large connectivity providers, such as NTT, which also host a lot of content. While they may be client-heavy, we only talk to their servers. The remaining 62% of client-heavy monitors see us because we respond to their scans. In the medium-visibility region, 100% of server-heavy monitors see us because they receive our client traffic. Client-heavy monitors still see us due to either their hosting of server content (48% of monitors) or scanning (43%). In two cases, a client-heavy monitor sees us due to peer-to-peer traffic. In low-visibility region, 100% server-heavy and 3% of client-heavy monitors see us due to our client traffic. Further, 19% of client-heavy monitors see us due

to peer-to-peer traffic, 47% due to their clients visiting our servers, and the rest due to scans they send to us, or we to them.

5.3 What We See of Others

Our analysis of who sees us (§ 5.2) is promising, but it allows us to only study who sees our client-heavy networks as targets. To get a broader view, we next flip this analysis around and examine what our networks see of targets out in the world. Our three universities provide client-heavy networks that see the world, and to these we add data from a large commercial CDN (*CDNX*) to evaluate what a server-heavy network sees of the world.

For targets, we begin with the 900 Random Far Monitors from § 5.2.2, and reuse their classification as client- or server-heavy. There are 875 unique /24-blocks when we eliminate duplicates. We map these 875 blocks to 295 ASes, since sometimes blocks are geographically-specific (for example, in a CDN). We call these ASes the *Random Far Targets*. We then consider all prefixes allocated to each of these ASes, making up about 5M /24-blocks. Using our universities and the CDN as monitors, we then study how many addresses and blocks are seen in each Random Far Target.

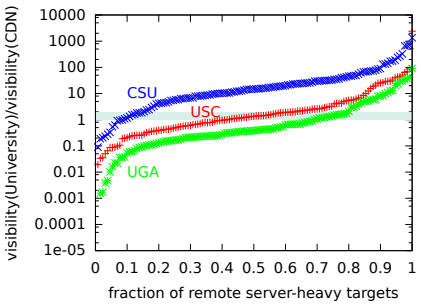
5.3.1 Correcting for sampling sparsity

Our datasets at universities may contain spoofed traffic. We correct for spoofing statistically as described in § 3.4.

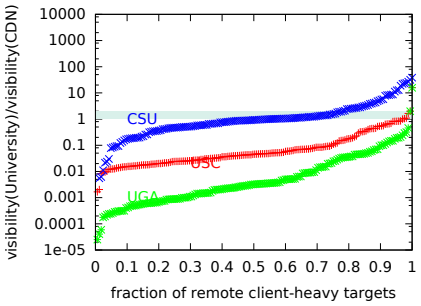
The second observation challenge is that our *CDNX* datasets both suffer from sampling sparsity. *CDNXglobal* covers all PoPs of the CDN, but only for one hour per day; it is sample-sparse in time. *CDNXpop* contains all data for the week, but it covers only two PoPs of the CDN (about 0.1% of their servers); it is sample-sparse in space. We use the *CDNXglobal* as our primary source, because the CDN shows intentional geographic bias at certain PoPs, causing spatially-sparse sampling to systematically under-represent visibility of many targets (it will miss targets that are geographically far from the PoP taking observations).

Let $\hat{V}_{m,t,T}$ be the visibility of the monitor m into target t for observation period T. Our goal is to find for all targets $\hat{V}_{allpop,*,1w}$, the estimated visibility from all of *CDNX* PoPs for a week-long observation, even though we only know $\hat{V}_{allpop,*,1h}$, the hourly visibility from everywhere in *CDNXglobal*, and $\hat{V}_{2pop,catch,1w}$, the weekly visibility at two PoPs in *CDNXpop* of some targets *catch*, which are mostly routed to those two PoPs, because they are geographically close. To correct *CDNXglobal*'s undercount due to sample-sparsity in time, we look for a scaling factor to estimate how much we miss by observing only one hour per day: $s^{catch}_{h\to w} = \hat{V}_{2pop,catch,1w}/\hat{V}_{allpop,catch,1h}$.

We estimate this scaling factor separately for addresses and for blocks. For blocks, $s^{b}_{h\to w} = 1.01$, so an hour-per-day at a single PoP sees nearly everything it would see in a week. For addresses, we need to know how much more will be observed *per block* instead of *in bulk*. Thus we estimate scaling factor for addresses per block in the PoP's catchment: $s^{a}_{h\to w}(b_i)$. Overall in 91% of blocks the scaling factor is lower than 2, and in about 45% of blocks the scaling factor is less than 1—*CDNXglobal* sees more of these prefixes than *CDNXpop*. These factors are expected, since CDNs cache popular content locally and most clients go to their nearby POPs. However, clients are sometimes routed to other PoPs for rarely used content. Such occasional routing leads to higher visibility of these client blocks in *CDNXglobal* than in *CDNXpop*. Overall this analysis supports our use of *CDNXglobal* as providing sufficient visibility, in spite of being temporally sparse. We adopt $s^{a}_{h\to w} = 2$, as this corrects sufficiently for 91% of blocks.



(a) server-heavy addresses



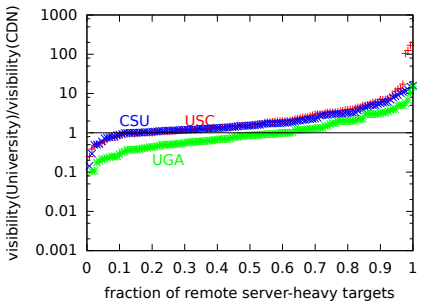
(b) client-heavy addresses

Figure 8: Visibility of addresses in server- and client-heavy targets, shown as the ratio of visibility of client-heavy monitors (at *USC*, *U.Ga.* and *CSU*) to an estimated visibility of server-heavy monitor (*CDNX*$_{cor}$).

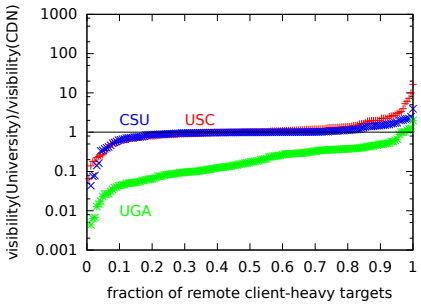
5.3.2 What client- and server-heavy networks see of others

After selection of the far targets and correction for our server-heavy monitor, we now compare what our three client-heavy monitors (at *USC*, *U.Ga.* and *CSU*) see relative to our corrected server-heavy monitor (*CDNX*$_{cor}$).

Figure 8 shows our results in observing addresses in two different types of targets. We see that our client-



(a) server-heavy /24 blocks



(b) client-heavy /24 blocks

Figure 9: Visibility of /24 blocks in server- and client-heavy targets, shown as the ratio of client-heavy monitors (at *USC*, *U.Ga.* and *CSU*) to an estimated server-heavy network ($CDNX_{cor}$).

heavy monitors (the three universities) generally see large numbers of servers: in Figure 8(a) *CSU* almost always sees more than $CDNX_{cor}$ (the blue line is above the horizontal line for $CDNX_{cor}$ for about 90% of targets), and *USC* sees more than $CDNX_{cor}$ for half of the targets. By contrast, these client-heavy universities see *less* of client-heavy targets: in Figure 8(b), universities as monitors almost always see less than $CDNX_{cor}$ (they are below the $y = 1$ line).

We see similar results when we count active blocks, as shown in Figure 9. The relative differences are smaller though (the graphs have different ranges), likely because there are more "mixed" blocks with both clients and servers (as was the case in our datasets, see Table 3), so there is more opportunity to see all blocks from any type of monitor.

We make two assumptions in this comparison: that we can correct for sampling sparsity in *CDNX* and that that the differences in sizes and activities at our targets do not dominate our comparison.

Conclusion: The trends in our data across hundreds of targets show that *WAN communication is bipartite, with client-heavy and server-heavy networks talking to each other more than within their own network class.* The implication is that *good overall visibility will require both powerful client* and *server monitors.*

6. OTHER KINDS OF SPARSITY

Although interest sparsity is the dominant factor, determining the overall coverage, care must be taken with other types of sparsity.

6.1 Temporal Sparsity

Temporal sparsity reflects the importance of listening "long enough". Prior work showed visibility increases logarithmically with time, with 70–90% of the addresses being discovered in the first 3 days [1, 6]. In each case, these prior studies evaluated effect of observation duration on the visibility that all their monitors had of the given targets (a university [1] and the Internet [6]). Inverted analysis allows us to study how temporal sparsity impacts *diverse* monitors.

We consider how visibility changes as we vary duration of passive observation – n – in unit of hours, from 1 up to the full duration. We start with all possible far monitors (all networks that see our three university targets), but discard very low-visibility monitors (80%), retaining only those that see the target in at least ten one-hour periods over the full trace. For space reasons, we report findings only for *USC* and only for addresses, but see similar results for our other two targets, and for blocks.

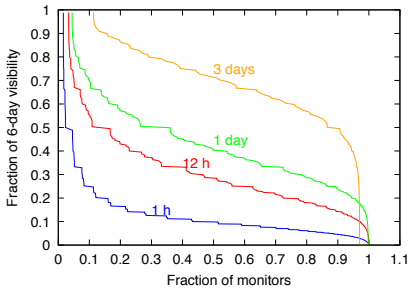
Figure 10 shows the distribution of visibility across all retained monitors as a function of time, compared to the baseline of evaluation over the full duration. Consistent with interest sparsity (Figure 4) we see a few monitors are able to reach their full coverage very quickly (5% of monitors need only one day). These are low-visibility monitors, whose coverage does not improve with time. After three days, 54% of monitors have seen at least 70% of their full visibility. These monitors' visibility exhibits logarithmic growth, with longer observations bringing reducing benefit. "Heavy see-ers" (§ 5.2.1), all reach 60–80% of their full visibility within a day, and reach more than 90% of full visibility within three days. Most monitors, however require multiple days and show linearly-increasing visibility even after 3 or 6 days. After three days (half of our observation period) 46% of monitors have achieved less than half of their maximum visibility.

Conclusion: Low-visibility monitors experience linear growth in their visibility and need longer observations to converge. Very low-visibility and medium to high-visibility monitors converge within days to at least 70% of their visibility.

6.2 Sampling Sparsity

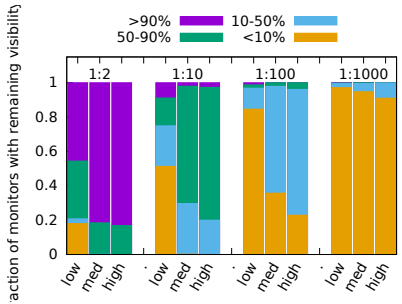
Sampling sparsity accounts for monitors that discard a fraction of traffic or flows, typically to keep up with a high-bitrate link (as in *USC*) or with limits on back-haul of monitor traffic (as in *CDNXglobal*).

To investigate sampling sparsity, we discard packets from flows with a given probability. If all packets are dis-

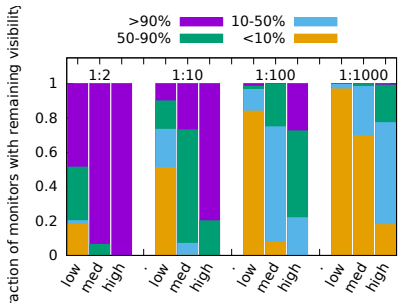


(a) Address visibility

Figure 10: Address visibility after one hour, twelve hours, one day and three days. (Block visibility is similar.)



(a) Address visibility



(b) Block visibility

Figure 11: Reduction in visibility of addresses and blocks when packets are sampled.

carded, we remove that flow. We then perform inverted analysis on the remaining flows, and compare visibility on non-sampled vs sampled flows.

Figure 11 shows the percentage of monitors in several ranges of remaining visibility—more than 90%, 50-90%, 10-50% and less than 10%. Monitors are grouped into low, medium and high visibility groups based on their visibility on non-sampled flows. Each group of bars shows a different sampling rate: 1 in 2, 10, 100, or 1,000.

We find that resiliency of monitors to sampling depends on their initial visibility. High- and medium-visibility monitors are barely affected by 1 in 2 sampling, and most just lose 50% of their address visibility with 1 in 10. Even at 1 in 100 sampling rate, these monitors retain much of their address visibility (between 10% and 50%).

(near) target	metric		far monitor *USC*	*U.Ga.*	*CSU*	*CDNX*
USC	addrs	near	-	125	14,495	17,673
		far	-	167	15,008	19,077
		near-cor.	-	158	14,541	18,549
	blks	near	-	84	357	388
		far	-	84	356	390
		near-cor.	-	80	350	386
U.Ga.	addrs	near	91	-	77	2,167
		far	262	-	575	3,574
		near-cor.	227	-	542	3.042
	blks	near	37	-	42	107
		far	158		80	202
		near-cor.	130		70	170
CSU	addrs	near	260	138	-	9,370
		far	308	161	-	14,881
		near-cor.	297	113	-	12,030
	blks	near	72	58	-	153
		far	88	81	-	203
		near-cor.	79	61	-	173

Table 5: Cross-validating sites to test coverage sparsity.

On the other hand, low-visibility monitors are severely affected by sampling. At 1 in 2 sampling, 60% of monitors lose half of their visibility.

Block visibility (Figure 11(b)) is much more robust to sampling than address visibility (Figure 11(a)), The opportunity to observe any addresses helps.

Conclusion: Sampling amplifies the effects of promiscuity, harming low-visibility monitors more than high-visibility ones.

6.3 Coverage Sparsity and Cross-Dataset Validation

Coverage sparsity is incompleteness in observing all traffic to the target. For example, in Figure 1, if the dotted link from k0/24 to the Internet exists, the near monitor will not see all traffic. We next evaluate coverage sparsity by comparing what each of our Known Networks see of each other. This comparison also serves as an end-to-end validation of our data sources.

To validate, we compare the visibility of each of our three targets by one of four far monitors (three university monitors and *CDNX*), as predicted by the inverted analysis, with the actual visibility calculated from the far monitor's observations. In effect, we use far as the ground truth against which we test near. If the near monitor misses addresses, that indicates either coverage sparsity (missing an incoming peering), or sampling sparsity (perhaps resulting from overload at the monitor, since we do not intentionally sample).

Table 5 summarizes these comparisons of our three universities against our four observers. We see good agreement between all sites about *USC*. However, remote sites see much more than predicted by our inverted analysis of *U.Ga.* and *CSU* (10–50%). (For example, *CDNX* sees 14 K addresses in *CSU*, while *CSU* expects them to see only 9 K addresses.) We believe our undercount at *CSU* occurs because of coverage sparsity—there are some blocks at CSU that are very active but only

seen by the far monitor, consistent with multi-homing. When we "correct" the target network by eliminating these never-monitored blocks (the near-cor. rows) we see much better agreement. At *U.Ga.*, we believe our undercount occurs because *U.Ga.* dataset is two years younger than the rest, so our predictions based on its current traffic, do not match the observations made two years prior.

7. RELATED WORK

Our work is motivated by the increasing use of passive sources to understand network services [1] and address liveness [7, 6, 28].

Early work compared passive and active techniques for discovering services in a campus network [1]. Bartlett et al. compared repeated active scans with a two-week long passive observation. They showed that duration of passive observation is critical, with popular servers appearing quickly; our evaluation of promiscuity generalizes and builds on this observation. Bartlett et al. also showed the role of third-party scanners in passive observation, and identified the difference in snapshot and continuous estimation of liveness due to dynamic addressing.

Dainotti et al. were the first to apply passive discovery to study Internet-wide liveness, complementing and expanding on active scanning [7, 6]. They recognize the importance of filtering spoofing, and identify the importance of multiple data sources. In their later work, they identify temporal and coverage factors that affect their passive observation, concluding from consistent results over time that neither factor biases their projected utilization [6]. We do not use their broad data sources and do not give new projections of Internet-wide address use. Instead, our work builds upon theirs to explore the root causes in the visibility provided by different monitors and the role of clients and servers. We also systematically study how visibility changes with sizes and types of monitors.

Zander et al. adopt the capture-recapture framework from estimation of biological populations, and apply it to extend prior passive and active estimates of Internet liveness [28]. They validate their approach on six chosen networks and use many data sources, however they do not explore the reasons their sources provide different information. Inspired by their work, we explore this question in § 4.5.

Overall, our work complements prior work by exploring the underlying reasons driving the visibility of passive sources, and sources of observation bias. We expect prior techniques can benefit from our analysis in source selection and to help understand and strengthen their findings.

8. CONCLUSION

In this paper we investigated what passive observers can learn about address liveness. We introduced the notion of sparsity to guide our understanding of when passive sources (monitors) add information, how much and why. We further developed inverted analysis as a new technique to allow evaluation of these questions by deriving many virtual monitors from a few actual monitors. We found that interest sparsity plays a key role in driving visibility, and that visibility is heavy-tailed. We also found that network traffic is mostly exchanged between client-heavy and server-heavy networks. Our insights can provide guidance to interpret existing evaluations of address liveness and to guide new measurements.

9. REFERENCES

[1] Bartlett, G., Heidemann, J., and Papadopoulos, C. Understanding passive and active service discovery. In *Proc. of ACM IMC* (San Diego, California, USA, Oct. 2007), ACM, pp. 57–70.

[2] Bloomberg. Example profile page. http://www.bloomberg.com/research/Stocks/private/snapshot.asp?privcapId=183695812.

[3] Borgnat, P., Dewaele, G., Fukuda, K., Abry, P., and Cho, K. Seven years and one day: Sketching the evolution of internet traffic. In *INFOCOM 2009, IEEE* (2009), IEEE, pp. 711–719.

[4] CAIDA. The caida anonymized internet traces 2011 dataset. http://www.caida.org/data/passive/passive_2011_dataset.xml.

[5] Chandy, K. M., and Lamport, L. Distributed snapshots: Determining global states of distributed systems. *ACM Transactions on Computer Systems 3*, 1 (Feb. 1985), 63–75.

[6] Dainotti, A., Benson, K., King, A., kc claffy, Glatz, E., Dimitropoulos, X., Richter, P., Finamore, A., and Snoeren, A. C. Lost in space: Improving inference of ipv4 address space utilization. *CoRR abs/1410.6858* (2014).

[7] Dainotti, A., Benson, K., King, A., kc claffy, Kallitsis, M., Glatz, E., and Dimitropoulos, X. Estimating Internet address space usage through passive measurements. *ACM Computer Communication Review 44*, 1 (Jan. 2014), 42–49.

[8] Dainotti, A., Benson, K., King, A., kc claffy, Kallitsis, M., Glatz, E., and Dimitropoulos, X. Estimating Internet address space usage through passive measurements. In *ACM Computer Communication Review* [7], pp. 42–49.

[9] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast internet-wide

scanning and its security applications. In *Proc. of USENIX Security Symposium* (Washington, DC, USA, Aug. 2013), USENIX, pp. 605–620.
[10] Fan, X., and Heidemann, J. Selecting representative IP addresses for Internet topology studies. In *Proc. of ACM IMC* (Melbourne, Australia, Nov. 2010), ACM, pp. 411–423.
[11] Heidemann, J. USC/LANDER passive and active data collection. Lightning talk at CAIDA AIMS Workshop, February 2009.
[12] Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., and Bannister, J. Census and survey of the visible Internet. In *Proc. of ACM IMC* (Vouliagmeni, Greece, Oct. 2008), ACM, pp. 169–182.
[13] Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., and Bannister, J. Census and survey of the visible internet. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement* (2008), ACM, pp. 169–182.
[14] IANA. Service Name and Transport Protocol Port Number Registry.
[15] IPInfo. Ipinfo. http://ipinfo.io.
[16] Liljenstam, M., Yuan, Y., Premore, B., and Nicol, D. A mixed abstraction level simulation model of large-scale internet worm infestations. In *Proc. of Tenth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (Oct. 2002), IEEE.
[17] LLC, Q. Argus: The network audit record generation and utilization system. http://qosient.com/argus.
[18] Mirkovic, J., and Kissel, E. Comparative evaluation of spoofing defenses. *IEEE Trans. Dependable Secur. Comput. 8*, 2 (Mar. 2011), 218–232.
[19] Moore, D., Shannon, C., Brown, D. J., Voelker, G. M., and Savage, S. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS) 24*, 2 (2006), 115–139.
[20] Neustar. Ip intelligence. https://ipintelligence.neustar.biz/.
[21] PeeringDB. PeeringDB. https://www.peeringdb.com/.
[22] Quan, L., Heidemann, J., and Pradkin, Y. Trinocular: Understanding Internet reliability through adaptive probing. In *Proc. of ACM SIGCOMM* (Hong Kong, China, Aug. 2013), ACM, pp. 255–266.
[23] Rivest, L.-P., and Baillargeo, S. Package rcaptuer. https://cran.r-project.org/web/packages/Rcapture/Rcapture.pdf.
[24] SANS. Internet storm center. http://dshield.org/.
[25] SPAMHAUS. The SPAMHAUS Project. http://www.spamhaus.org/.
[26] Views, R. University of oregon route views project, 2000.
[27] Xu, J., Fan, J., Ammar, M. H., and Moon, S. B. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *Proc. of 10thIEEE International Conference on Network Protocols* (Washington, DC, USA, Nov. 2002), IEEE, pp. 280–289.
[28] Zander, S., Andrew, L. L., and Armitage, G. Capturing ghosts: predicting the used ipv4 space by inferring unobserved addresses. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014), ACM, pp. 319–332.

Acknowledgments

The authors would like to thank Christos Papadopoulos, Roberto Perdisci and Yuri Pradkin for their help with data collection and processing. We are grateful to Colorado State University, University Of Georgia, University of Southern California and our anonymous CDN for providing data to support this work.

John Heidemann's work is partially sponsored by the U.S. Dept. of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, via SPAWAR Systems Center Pacific under Contract No. N66001-13-C-3001, and via BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement numbers FA8750-12-2-0344 and FA8750-15-2-0224. The U.S. Government is authorized to make reprints for Governmental purposes notwithstanding any copyright. The views contained in herein are those of the authors and do not necessarily represent those of DHS or the U.S. Government.

APPENDIX

A. THE ROLE OF UNSOLICITED TRAFFIC

In § 5 we explored the role of how bipartite traffic between client- and server-heavy networks affects visibility. In this appendix we expand on that analysis and consider how unsolicited traffic, such as scans and backscatter, affects visibility. Prior work has shown that scanners can make large contributions to passive observation at a University [1] target, helping fast discovery of *servers*. Our results here complement these findings in the following ways:

1. We confirm that unsolicited traffic is crucial for server discovery, and that it can also be used to effectively discover clients (§ A.1)

2. Most monitors do not gain visibility through scans and most monitors do not participate in wide scanning. Other types of unsolicited traffic, such as backscatter or unsuccessful connection attempts, provide greater visibility than scanning (§ A.2).
3. The effectiveness of scans at discovering hosts or services depends greatly on the target, as well as the type of scan (for example, ICMP or TCP SYN) (§ A.3).
4. Local policies affect visibility (§ A.4) of a target.

A.1 Unsolicited Traffic May Lead to Good Visibility

Our first result is that many addresses and blocks are visible through unsolicited traffic. We define useful traffic as traffic where we can identify the client and the server, and show that both parties actively participate in the conversation—there is payload exchanged in both direction of the connection. On the other hand, unsolicited traffic consists of scans or unsuccessful connection attempts, backscatter, and ICMP probes (ECHO packets and replies). Table 2 shows all flow labels, which are grouped under "unsolicited" category.

Table 6 shows the percentage of addresses and blocks at each target that are visible through either useful or unsolicited traffic, for all virtual monitors. This figure also breaks out visibility by address type, as defined in § 3.3.2 and Table 3.

While useful traffic reveals many blocks (useful traffic contributes to visibility of 78–86% of addresses and 82–91% of blocks), we see that unsolicited traffic shows even more (89–99% of addresses and 95–98% of blocks). This improvement occurs because of two phenomena. First, unsolicited traffic into our targets probes all addresses—some of these probes receive a reply from addresses that are not otherwise active. Second, some addresses in our targets send only unsolicited traffic, and appear to be infected by Trojans.

A.2 High Visibility Usually Comes from Useful Traffic

Our second result is that most high-visibility monitors achieve their high visibility through useful traffic. We again regard traffic as useful or unsolicited, using our flow classification from § 3.3.2 (Table 2).

We expect web traffic to be the dominant contributor in useful traffic, and demonstrate this is the case by singling out web traffic based on port usage (identifying web as TCP ports 80, 443, and 8080).

Finally, we label each flow with regard to the target address's role in this flow as "src" or "dst". For example, a label "useful-src" identifies a useful flow originated by the target toward some server. Unfortunately, as described in § 3.3.1, Argus reverses the source and destination addresses for some ICMP flows in a way that we cannot identify and correct. As a result, we cannot guarantee correct identification of the source or destination of unsolicited ICMP traffic. This deficiency may artificially skew the results for unsolicited traffic when broken down by origin, elevating counts for the "unsolicited-src" category.

Figure 12 shows contributions of different classes of traffic (useful, web-src, web-dst, unsolicited-src and unsolicited-dst) to overall visibility for each virtual monitor. The sum of all contributions exceeds 1, since any given address could be observed through more than one traffic class. To smooth the noise, we average the contributions for groups of monitors whose visibility is within 1% of each other. Each sub-figure shows one of our targets, with the x-axis showing the address visibility of virtual monitors.

When we consider virtual monitors with high visibility (the left side of each graph), we see that for most monitors, high visibility comes from the presence of *useful* traffic (the dark purple line is much higher than the orange and yellow lines on the left side of each graph). Web traffic sent from our targets to servers in virtual monitor networks (the green line) shows an especially large difference. This data shows that, while some high-visibility monitors see a lot of our targets through scanning, it is much more common for monitors to see our targets because they host popular content that is accessed by clients in the target. Conversely, external monitors that see 0.01% to 1% of the target often see it through unsolicited traffic. This finding holds across all three targets.

A.3 Visibility through Direct Scans Depends on Scan Type and the Target

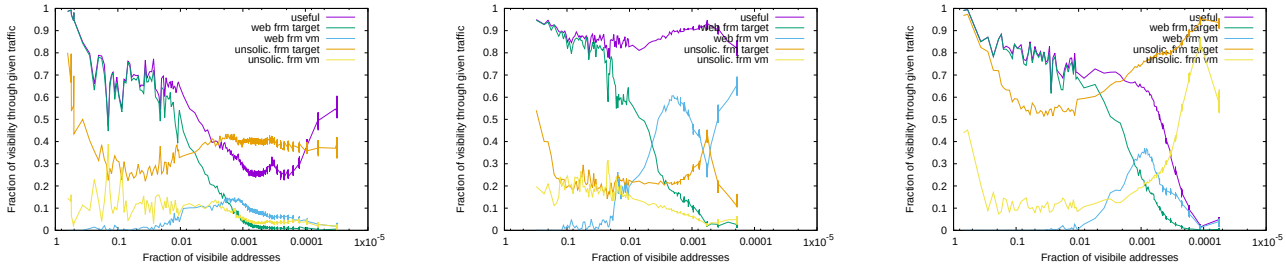
Our third finding is that visibility through direct scans to the target may sometimes be quite low, and that different protocols used in scans dominate visibility for different targets.

We define a direct scan that receives a reply as: (1) ICMP echo packet that receives ICMP echo reply, (2) TCP SYN going from a non-service port to a service port that receives TCP SYN-ACK or TCP RST, (3) UDP packet going from a non-service port to a service port that receives a reply from that port or that leads to an ICMP unreachable reply. We break out visibility through unsolicited traffic, into visibility afforded by direct scans to the target, which receive a reply, and visibility through other unsolicited traffic. We then subdivide direct scans by protocol into scan/ICMP (ICMP echo request or reply) and scan/non-ICMP (that is, TCP and UDP).

Table 7 shows that these subclasses of traffic make very different contributions across our targets. There is no clear trend: for *USC*, ICMP and non-ICMP discover similar fractions of addresses and blocks. For *CSU*, non-ICMP discover much more than ICMP. Finally, no ICMP appears in our *U.Ga.* data, and non-ICMP gets visibility

traffic category by target	client-only	server-only	client-server	ambiguous	prober	responder	one-way source	addresses	blocks
useful									
USC	98%	88%	97%	0%	0%	0%	0%	85%	82%
UGA	98%	94%	99%	0%	0%	0%	0%	86%	91%
CSU	95%	35%	90%	0%	0%	0%	0%	78%	90%
unsolicited									
USC	97%	99%	99%	96%	100%	100%	100%	98%	98%
UGA	91%	68%	98%	41%	0%	100%	100%	89%	95%
CSU	99%	99%	100%	88%	100%	100%	100%	99%	95%

Table 6: Visibility of various address categories through different traffic types



(a) USC (b) UGA (c) CSU

Figure 12: Contributions of different classes of traffic to address visibility: virtual monitors are grouped by their power.

similar to what it does at *CSU*. We conclude that *site-specific policies and data sources* affect visibility.

Data to support this claim of variation comes from comparing Table 6 and Table 7. Comparing numbers for visibility through unsolicited traffic in Table 6 (last two columns) and visibility through direct scans in (last column), we see that 26–67% of addresses and 7–10% of blocks are visible through the unsolicited traffic other than replies to scans. For example, unsolicited traffic reveals 98% of addresses and blocks at *USC*, but only 72% of addresses and 91% of blocks are revealed due to their replies to direct scans. The rest are revealed because they either reply to backscatter traffic, or they have some failed attempts to initiate connections to popular servers. Thus direct scans have limited success at discovering live addresses. This limited success may result from different causes: either a network may identify scanners and filter out their traffic, or addresses may be dynamic and may be scanned at the time when they are not assigned to a host. We investigate the first reason in the next section, and leave the second for future work.

A.4 Local Policies Affect Visibility

Our fourth finding is that *local* security policies greatly affect visibility achieved through scanning. Our data indicates that *U.Ga.* and *CSU* employ some scan filtering, which greatly reduces their visibility through direct scans. Looking at Table 7 almost all allocated *USC*'s addresses and blocks receive both non-ICMP and ICMP scans. Further, 53% of live addresses (90% of blocks) reply to non-ICMP scans, and 58% of live addresses

Scan type	Scanned (% alloc.)	Replied (% live)		
		Posit.	Negat.	All
USC				
non-ICMP	99% (99%)	31% (75%)	22% (87%)	53% (90%)
ICMP	99% (99%)	58% (89%)	0 (0)	58% (89%)
all	99% (99%)	66% (90%)	22% (87%)	72% (91%)
UGA				
non-ICMP	11% (82%)	23% (81%)	16% (60%)	30% (85%)
ICMP	0 (0)	0 (0)	0 (0)	0 (0)
all	11% (82%)	23% (81%)	16% (60%)	30% (85%)
CSU				
non-ICMP	8% (62%)	11% (61%)	22% (85%)	30% (85%)
CMP	66% (68%)	6% (55%)	0 (0)	6% (55%)
all	66% (69%)	14% (74%)	22% (85%)	32% (85%)

Table 7: Percentages of addresses and blocks per target that receive direct scans and that reply to scans.

(89% of blocks) reply to ICMP scans. *U.Ga.* traces contain only TCP and UDP traffic. Only 11% of allocated *U.Ga.* addresses (82% of blocks) receive non-ICMP scans. Because most of blocks are scanned, and because the smallest routing unit is usually a /24 block, we believe that our monitor sees all incoming traffic for allocated *U.Ga.* blocks. We also have no reason to believe that *U.Ga.* addresses are less attractive to scanners. Instead, we hypothesize that the small percentage of addresses scanned points to an aggressive scan filtering policy at *U.Ga.*. Out of live *U.Ga.* addresses and blocks, 30% of addresses and 85% of blocks reply to non-ICMP scans.

With regard to allocated address space at *CSU*, 8% of addresses (62% of blocks) receive non-ICMP scans, and 66% of addresses (68% of blocks) receive ICMP scans. Again, because majority of blocks receive both types of scans, we conclude that the small percentage

of probed addresses points to aggressive scan filtering policy. Out of live *CSU* addresses and blocks, 30% of addresses (85% of blocks) reply to non-ICMP scans, but only 6% of addresses (55% of blocks) reply to ICMP scans. This points to further filtering of ICMP scans or replies, either at the host or at the network level.