Collaborative Estimation of the Size of the Used IPv4 and IPv6 Address Spaces

Sebastian Zander, Lachlan L. H. Andrew, Grenville Armitage
Centre for Advanced Internet Architectures, Technical Report 130930B
Swinburne University of Technology
Melbourne, Australia
szander@swin.edu.au, landrew@swin.edu.au, garmitage@swin.edu.au

***Abstract*—In order to better understand how the transition from IPv4 to IPv6 will play out, we need to know how much of the allocated IPv4 space is *actively used* and how many hosts *actually use* IPv6. This report describes a collaborative, secure, anonymised scheme for estimating IPv4 and IPv6 address space utilisation based on private datasets of locally observed IP addresses, such as server logs or traffic traces. We are looking for collaborators willing to participate.**

I. Introduction

We are looking for collaborators willing to participate in a secure, anonymised scheme for estimating IPv4 and IPv6 address space utilisation based on private datasets of locally observed IP addresses (e.g. server logs or traffic traces) and the capture-recapture (CR) approach [1]. Our efficient private set intersection cardinality (PSIC) protocol allows collaborators to contribute to this Internet-wide study while ensuring anonymity of the IP addresses each collaborator has observed [2], [3].

As of mid 2013 more than 95% of the usable IPv4 address space has been allocated and according to predictions, the Regional Internet Registrars (RIRs), except AfriNIC, will run out of IPv4 addresses by the end of 2014 [4]. However, the transition to IPv6 is still relatively slow. Some larger service providers support IPv6, but many smaller service providers do not support IPv6 yet [5]. Also, only a few percent of clients are capable of using IPv6 without 6to4 or Teredo tunelling [6].

While most of the IPv4 address space has been *allocated*, it is unclear how many allocated addresses are *actively used* (simply referred to as *used*). Knowing how many addresses are used is important to predict the value and costs of an IPv4 address market. Over 30 IPv4 address sales have been reported already [7]. Also, once the IPv4 space is fully allocated, its progressive exhaustion can only be measured through tracking the usage. Furthermore, tracking the used IPv4 address space provides insights into the IPv6 deployment time frame: the fewer unused IPv4 addresses remain, the higher the pressure is to adopt IPv6. On the other hand, tracking the used IPv6 addresses provides important insights into how many hosts actually use IPv6.

Until recently little research was published on identifying how much of the IPv4 space is used. The previous studies were based mostly on active probing ("pinging") of the IPv4 address space, so they likely underestimated the actually used space [8]–[11]. Furthermore, the existing studies are outdated, with the exception of [11], which is based on pinging alone. More research exists on the deployment progress of IPv6 as outlined in [1], [5], [12], but many studies only analysed certain sub populations, e.g. their analysis is based on a few particular server/traffic logs, or they analysed the IPv6 capabilities of either public servers or clients.

We developed a technique [1] that combines several different data sources of observed used IPv4 addresses (e.g. server logs or traffic traces) and uses CR [13]–[16] to estimate the total population of used addresses, including the *unobserved* used addresses. The same technique could be applied to estimate the used IPv6 space. However, a diverse set of data sources is required to get a good "coverage" of the IP address space and produce a good CR estimate. For CR we need to know the number of used IP addresses observed 1) by each data source and 2) by all combinations of set intersections of all sources.

Many data sources of used IP addresses exist. The challenge is to efficiently combine the data sources of multiple collaborators in a secure manner. We assume the number of addresses observed by one source is not sensitive information, but the observed IP addresses must be kept confidential. We developed an efficient and secure PSIC protocol that computes the intersection cardinalities between data sources needed for CR while *ensuring the anonymity* of the IP addresses observed.

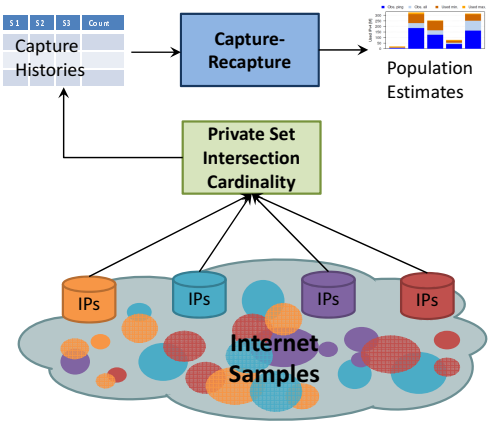

Figure 1. Different data sources sample different partially-overlapping parts of the Internet; with a private set intersection cardinality protocol we can securely compute how many addresses each combination of sources captured (capture histories), and then use a capture-recapture method to estimate the population.

We are looking for *more collaborators* willing to participate in our scheme summarised in Figure 1. Collaborators can contribute unanonymised or anonymised data and work with us on refining the estimation method. Participating in this effort has the benefits of getting timely information about the state of the IPv4 and IPv6 address utilisation of the Internet, publishing joint research, and making a meaningful and positive impact by supporting the whole Internet community.

In Section II we describe our overall approach in more detail. In Section III we discuss how to estimate the population with CR. In Section IV we discuss how to compute the input data for CR while ensuring the anonymity of addresses with PSIC. A prototype implementation of our proposed PSIC protocol is publicly available [3].

II. Overall Approach

Our efficient PSIC protocol allows collaborators to contribute observed IP address data while ensuring anonymity of the IP addresses each collaborator has observed. The CR approach [1] allows to estimate the IPv4 and IPv6 address space utilisation based on address data samples. Figure 1 shows an overview of the PSIC+CR approach. We now explain this approach in more detail.

Several collaborators have access to data sources of observed used IPv4 or IPv6 addresses. Data sources can be server logs, traffic traces, or data from active probing. Each data source is an incomplete sample of the whole used space (total population). The different samples may be "biased", e.g. towards certain geographical areas or certain types of hosts. However, with a possibly very small but non-zero probability one IP address will appear

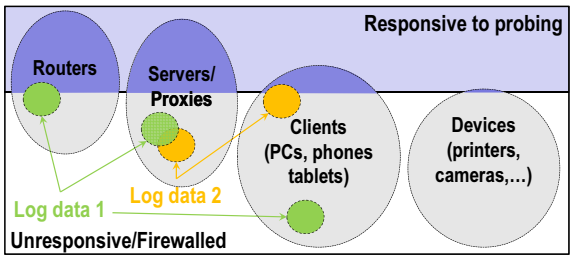

Figure 2. Different addresses have different visibility depending on the types of hosts

in any of the data sources. We assume that the IP addresses collected were actually used. This is usually the case if addresses are collected from server logs (where the application is TCP-based, such as web traffic) or via active probing. In other cases (e.g. for traffic traces) it may be necessary to filter out potentially spoofed IP addresses.

Different types of hosts have different chances of appearing in a data source (see Figure 2). We differentiate between the following types: routers, servers/proxies, general-purpose clients (e.g. PCs, mobile devices), and specialised devices (e.g. printers, IP phones, cameras). Many routers or public servers respond to active probing. Server logs are a good source of addresses of certain sets of clients, but possible also log specialised devices or servers (e.g. operating system update servers). Traffic traces may contain addresses of all host types. Devices that only communicate in private networks are effectively invisible. However, we assume such devices are relatively rare, as most devices use at least some communication, e.g. to update their software or their clocks.

The heterogeneity of the observed IP addresses due to different factors, such as hosts types or geographic dependencies, is the reason we need diverse data sources from multiple collaborators. Only then can we achieve sufficient "coverage". We assume the collaborators are willing to share the sizes of their data sources, i.e. how many IP addresses each data source contains. However, many collaborators may not want to share their observed IP addresses. Hence, all collaborators run the PSIC protocol to securely compute the intersection cardinalities between all combinations of data sources.

With the sizes of the data sources and the sizes of all combinations of intersections between datasets we can compute the table of "capture histories". This table contains the number of addresses observed by each combination of sources, for example the number of addresses observed only by source 1, the number of addresses observed only by source 2, the number of addresses

observed by source 1 and source 2, and so on (see Table I). Based on the capture histories CR techniques can estimate the number of addresses not observed by any of the sources and hence the total population.

The PSIC+CR scheme can be run periodically, each time with data from the last time window, to estimate the population trend over time. Furthermore, the data sources can be stratified by all collaborators and population estimates can be computed separately for each stratum. For example, IP addresses can be grouped by the RIR responsible for their allocation to detect broad geographical trends. Using PSIC+CR with stratified data leaks information about the datasets, but the leakage may be tolerable if there are only a few coarse strata.

III. Capture-Recapture Method

The simplest CR model is the two-sample Lincoln-Petersen (L-P) method [13]–[14], which works as follows. Given a first sample, of M individuals, the size of the population would be known if we knew what *fraction* of the population had been sampled. To estimate this, L-P takes a second sample of C individuals, of which R individuals occur in both samples. If the fraction of "recaptured" individuals in the second sample equals the fraction of the total population captured in the first sample, then the population N can be estimated by [13]–[14]:

$$R/C = M/N\ , \quad N = \frac{MC}{R}\ .$$

In the context of Internet address estimation, the samples or "sources" are different active and passive measurements and the individuals are IPv4 or IPv6 addresses. For concreteness, consider one source to be all addresses that responded to pinging of the entire IPv4 space and another to be all addresses in a traffic trace.

The L-P estimate assumes that the probability of an individual being captured in one source does not depend on the probability of being captured in a different source (*independent sources*). It also assumes that, within a sample, each individual has an equal chance of being sampled (*homogenous population*), specifically that the probability is not zero for any individual. Individuals with zero sample probability are not part of the estimated population. Furthermore, the L-P estimate assumes that during measurement no individuals enter or leave the population (*closed population*), but a violation of this assumption is simply another form of heterogeneity.

However, there are CR models that can cope with heterogeneity and/or source dependence, for example log-linear models [15]–[17]. These methods require more than two data sources and knowledge of the "capture histories" of individuals.

Table I
Example three-source capture history table

Source 1	Source 2	Source 3	Count
0	**0**	**0**	Z_{000} =?
0	**0**	**1**	Z_{001}
0	**1**	**0**	Z_{010}
0	**1**	**1**	Z_{011}
1	**0**	**0**	Z_{100}
1	**0**	**1**	Z_{101}
1	**1**	**0**	Z_{110}
1	**1**	**1**	Z_{111}

Let N be the unknown number of distinct individuals of the population. Let t denote the number of sources indexed by $1, 2, \ldots, t$. For each individual, let s_1 to s_t be defined such that $s_i = 1$ if the individual occurs in sample i and $s_i = 0$ otherwise. Then the string $s_1 s_2 \ldots s_t$ is called the "capture history" of the individual. The observed outcome of all measurements can then be represented by variables of the form z_s, which are the numbers of individuals with each capture history $s = s_1 s_2 \ldots s_t$. These are assumed to be instances of random variables Z_s.

Note that individuals with the capture history $00\ldots0$ are unobserved, and our goal is to estimate $Z_{00\ldots0}$. If $M = \sum_{S\setminus\{00\ldots0\}} Z_s$ is the total number of observed individuals, then the estimated population size is $\hat{N}_P = M + \hat{Z}_{00\ldots0}$. While collaborators often cannot share the lists of individuals for privacy reasons, we assume they can share the number of individuals in a source N_i. Then our PSIC protocol described in Section IV can be used to compute all Z_s other than $Z_{00\ldots0}$ based on the intersection cardinalities of combinations of sources and the known N_i.

For $t = 3$ there are seven known capture counts $Z_{001}, Z_{010}, \ldots, Z_{111}$ as shown in Table I. For example, Z_{111} is the number of individuals captured by source 1, source 2 and source 3, so Z_{111} is computed as intersection cardinality of source 1, source 2 and source 3. To compute the counts of individuals in only one source i we need N_i. For example, Z_{001} is the number of individuals only in source 3 and is computed as $Z_{001} = N_3 - Z_{011} - Z_{101} - Z_{111}$.

We now briefly explain how to use log-linear models to estimate $Z_{00\ldots0}$ (for details see [1]). For each history s, let $h(s)$ be the set of samples in which the individual occurs; for example, $h(101) = \{1, 3\}$. Define the indicator function $\mathbf{1}_A = 1$ if statement A is true and 0 otherwise.

We can now write the following system of equations in 2^t variables $u, u_1, u_2, \ldots, u_{12}, \ldots, u_{23}, \ldots$ up to $u_{12\ldots t}$:

$$\log(\mathbb{E}(Z_s)) = \sum_{h \subseteq h(s)} u_h = \sum_h u_h \mathbf{1}_{h \subseteq h(s)} .$$

For example, for $t = 3$, the system is

$$\begin{aligned} \log\left(\mathbb{E}\left(Z_{ijk}\right)\right) =& u + u_1 \mathbf{1}_{i=1} + u_2 \mathbf{1}_{j=1} + u_3 \mathbf{1}_{k=1} \\ &+ u_{12} \mathbf{1}_{i=1 \wedge j=1} + u_{13} \mathbf{1}_{i=1 \wedge k=1} \\ &+ u_{23} \mathbf{1}_{j=1 \wedge k=1} + u_{123} \mathbf{1}_{i=1 \wedge j=1 \wedge k=1} . \end{aligned}$$

The estimate of $Z_{00\ldots0}$ is then $\hat{Z}_{00\ldots0} = \exp(u)$. If we take $\mathbb{E}[Z_s] = z_s$ then this system has 2^t unknowns but only $2^t - 1$ equations, as $Z_{00\ldots0}$ is unknown. Hence it is customary to assume $u_{12\ldots t} = 0$ [15]. Using all u_h usually results in over-fitting. Model selection techniques are used to select the least complex model with "adequate" fit; effectively some u_h are forced to 0, to reflect assumed independence between certain combinations of sources [1].

Note that with the capture counts Z_s estimates for $\hat{Z}_{00\ldots0}$ and $\hat{N}_P$ can be computed using *many* existing CR methods, not only log-linear models.

IV. Private Set Intersection Cardinality Protocol

We assume that all collaborators participating in the PSIC protocol are potential honest-but-curious adversaries; they run the protocol correctly, but try to learn as much information as possible. We assume that man-in-the-middle attacks by third parties are prevented with properly configured SSL/TLS connections. Our protocol works with two or more parties (but we expect no more than 10–20 parties), and it does not need trusted third parties. Our protocol is based on well proven techniques and is reasonably efficient.

Let k be the number of parties. Each party i has a dataset D_i with $N_i = |D_i|$ distinct addresses and a private encryption key K_i. Our protocol is based on the approach proposed in [18], [19], which is based on *commutative encryption*. With commutative encryption different datasets can be encrypted IP address by IP address, and identical addresses in the original datasets will always result in identical encrypted addresses (ciphertexts) in datasets encrypted by all parties (*fully-encrypted* datasets), regardless of the order in which the parties encrypted the datasets.

Pohlig-Hellman (PH) encryption is a secure commutative encryption scheme [20]. The encryption function is:

$$E_{K_i}(m) = m^{K_i} \pmod{p} , \qquad (1)$$

where m is the plaintext message (an IP address), p is a large safe prime number[1] shared by all parties and $\gcd(K_i, p-1) = 1$. It is easy to see that this function is commutative since

$$\begin{aligned} E_{K_i}\left(E_{K_j}(m)\right) &= m^{K_i K_j} \pmod{p} \\ &= m^{K_j K_i} \pmod{p} = E_{K_j}(E_{K_i}(m)) . \end{aligned}$$

The protocol works as follows (for a detailed description see [2]). Each party encrypts the IP addresses of their own dataset using their own private key, randomly permutes the encrypted addresses, and then passes the encrypted and permuted dataset to the next party – a party that has not encrypted and permuted this dataset before. The next party encrypts and randomly permutes the received dataset with their own private key, passes it to the next party that has not had this dataset and so on, until all datasets are fully-encrypted. Finally, the parties share the fully-encrypted datasets among each other. Since the encryption is commutative, the set intersection cardinality of the ciphertexts is identical to the set intersection cardinality of the plaintexts.

The scheme is computationally secure, since no party can decrypt any of the other parties' datasets without knowing the other parties' encryption keys and because of the random permutations no party knows which ciphertexts map to which plaintexts.

Since the scheme encrypts each IP address of a dataset separately, the space overhead is very large. For example, with a modulus of 1024 bits (NIST 2010 Legacy [21]) each encrypted 4-byte IPv4 address results in a 128 bytes long ciphertext. Also, the computational overhead is high given the exponentiation-based commutative encryption function (Equation 1). For large datasets the scheme is impractical.

To make the scheme practical we propose that prior to running the PSIC protocol all parties generate sampled IP address datasets of much smaller size. Hash-based sampling can be used to sample addresses consistently across all parties. We can sample the same IP addresses from different data sources by selecting only the addresses with

$$H(m + s) \pmod{R} < r ,$$

where m is the hash input (an IP address), s is a sampling salt and r/R is the sampling rate ($r, R \in \mathbb{N}$). The sampling rate can take any value, since r/R can be

[1] A safe prime is a prime of the form $2p + 1$, where p is a prime.

any rational number in $(0, 1]$. All parties need to agree on H, s, r and R. The salt s randomises the choice of sample and could be computed in a shared fashion, e.g. each party contributes some bits. The actual intersection cardinality can be estimated based on the intersection cardinality of the sampled datasets [2].

The above scheme is not resistant to probing attacks – attacks where one party generates datasets with mostly invalid addresses to test whether a few valid addresses are in another party's dataset. Since no party can decrypt the fully-encrypted datasets it is impossible to check whether an original dataset contained only valid addresses. We developed a novel, optional mechanism that allows to detect and prevent probing attacks.

Prior to encrypting the actual datasets, all parties agree on the set of valid addresses (*valid set*). For example, in the context of IPv4 addresses the valid set could be the set of advertised and routed IPv4 addresses (e.g. based on Routeviews [22]). Then the parties essentially perform PSIC as outlined above with the valid sets. Each party encrypts each other party's valid set using the *same* encryption key used later to encrypt the datasets. By doing this each party obtains a fully-encrypted valid set, which can be used later to check if a fully-encrypted dataset contains mostly valid addresses.

If the set intersection cardinality between a fully-encrypted dataset and a fully-encrypted valid set is under some threshold, a probing attack is detected [2]. Not returning the fully-encrypted probe dataset to the prober prevents the attack. To make the probing prevention scalable the valid sets can be sampled with a low sampling rate using hash-based sampling [2].

Acknowledgements

This research was supported under Australian Research Council's Linkage Projects funding scheme (project LP110100240) in conjunction with APNIC Pty Ltd and by Australian Research Council grant FT0991594.

References

[1] S. Zander, L. L. H. Andrew, G. Armitage, G. Huston, "Estimating IPv4 Address Space Usage with Capture-Recapture," in *(accepted at) 7th IEEE Workshop on Network Measurements (WNM) in conjunction with IEEE LCN*, October 2013.

[2] S. Zander, L. L. H. Andrew, G. Armitage, "Scalable Private Set Intersection Cardinality for Capture-Recapture with Multiple Private Datasets," Tech. Rep. 130930A, Centre for Advanced Internet Architectures, Swinburne University of Technology, 2013.

[3] S. Zander, "Secure Fast Set Intersection (SeFaSI) Implementation," 2013. http://caia.swin.edu.au/sting/sefasi.

[4] G. Huston, "IPv4 Address Report." http://www.potaroo.net/tools/ipv4/index.html.

[5] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, M. Bailey, "Measuring IPv6 Adoption," Tech. Rep. TR-13-004, International Computer Science Institute, August 2013. http://www.icir.org/mallman/papers/icsi-tr-13-004.pdf.

[6] S. Zander, L. L. H. Andrew, G. Armitage, G. Huston, G. Michaelson, "Mitigating Sampling Error when Measuring Internet Client IPv6 Capabilities," in *ACM Internet Measurement Conference (IMC)*, Nov. 2012.

[7] IPv4 Market Group, "Recent Sales." http://ipv4marketgroup.com/home/, viewed 26th August 2013.

[8] Y. Pryadkin, R. Lindell, J. Bannister, R. Govindan, "An Empirical Evaluation of IP Address Space Occupancy," Technical Report ISI-TR 598, USC/ISI, 2004.

[9] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, J. Bannister, "Census and Survey of the Visible Internet," in *ACM Conference on Internet measurement (IMC)*, pp. 169–182, 2008.

[10] X. Cai, J. Heidemann, "Understanding Block-level Address Usage in the Visible Internet," in *ACM SIGCOMM Conference*, pp. 99–110, 2010.

[11] Anonymous, "Internet Census 2012 – Port scanning /0 using insecure embedded devices." http://internetcensus2012.bitbucket.org/paper.html.

[12] kc claffy, "Tracking IPv6 Evolution: Data We Have and Data We Need," *ACM Computer Communication Review (CCR)*, vol. 43, pp. 43–48, July 2011.

[13] C. G. J. Petersen, "The Yearly Immigration of Young Plaice into the Limfjord from the German Sea," *Rept. Danish Biol. Sta.*, vol. 6, pp. 1–77, 1895.

[14] F. C. Lincoln, "Calculating Waterfowl Abundance on the Basis of Banding Returns," *U.S. Dept . Agric. Circ.*, vol. 118, pp. 1–4, 1930.

[15] E. B. Hook, R. R. Regal, "Capture-Recapture Methods in Epidemiology: Methods and Limitations," *Epidemiol. Rev.*, vol. 17, no. 2, pp. 243–264, 1995.

[16] A. Chao, "An Overview of Closed Capture-Recapture Models," *J. Agric. Biol. Envir. S.*, vol. 6, no. 2, pp. 158–175, 2001.

[17] A. Chao, P. K. Tsay, S. H. Lin, W. Y. Shau, D. Y. Chao, "The Applications of Capture-Recapture Models to Epidemiological Data," *Statistics in Medicine*, vol. 20, pp. 3123–3157, October 2001.

[18] R. Agrawal, A. Evfimievski, R. Srikant, "Information Sharing Across Private Databases," in *ACM SIGMOD International Conference on Management of Data*, pp. 86–97, 2003.

[19] J. Vaidya, C. Clifton, "Secure Set Intersection Cardinality with Application to Association Rule Mining," *J. Comput. Secur.*, vol. 13, pp. 593–622, July 2005.

[20] S. Pohlig, M. Hellman, "An improved algorithm for computing logarithms over and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. 24, pp. 106–110, January 1978.

[21] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, "Recommendation for Key Management." Special Publication 800-57 Part 1 Rev. 3, NIST, July 2012.

[22] University of Oregon Route Views Project. http://www.routeviews.org/.