Observations of IPv6 Addresses

David Malone <David.Malone@nuim.ie>

Hamilton Institute, NUI Maynooth.*

Abstract. IPv6 addresses are longer than IPv4 addresses, and are so capable of greater expression. Given an IPv6 address, conventions and standards allow us to draw conclusions about how IPv6 is being used on the node with that address. We show a technique for analysing IPv6 addresses and apply it to a number of datasets. The datasets include addresses seen at a busy mirror server, at an IPv6-enabled TLD DNS server and when running traceroute across the production IPv6 network. The technique quantifies differences in these datasets that we intuitively expect, and shows that IPv6 is being used in different ways by different groups.

1 Introduction

IPv6 uses an address space that is far larger than could be consumed by devices in the near future. The reason for such a large address space is to try to make address management easier, both when numbering hosts within subnets and when numbering networks within the Internet. The hope is that addresses and subnets can be assigned according to logical schemes rather than assigning addresses in the most compact way.

This is in contrast to allocation of addresses for IPv4; in the CIDR world it is unclear where the network address ends and the host address begins. Some deductions can be made about IPv4 addresses: we can consult IANA and whois databases to determine who addresses have been assigned to. However, unless whois/DNS information is well maintained, it will be difficult to know how addresses are actually being used.

In this paper, we show how to use IPv6's extra expressiveness to infer things about how addresses are being used. As with IPv4, we can consult IANA in order to discover which registrar has been assigned the address. However, we can also identify people who connect to the IPv6 Internet using mechanisms such as 6to4 [5] and Teredo [7]. As there are standard procedures for allocating host IDs, we can identify auto-configured hosts and other address schemes in use.

Such an analysis of IPv6 address is not difficult in itself; a competent IPv6 network engineer could perform this analysis by glancing at an address. However, we will automate this analysis and apply it to large datasets. The first dataset is the IPv6 addresses observed in the wild at `ftp.heanet.ie`. Our second is the set of recursive DNS servers making queries to `ns6.iedr.ie`, an authoritative server for the *ie* domain. Our third is based on addresses responding to a traceroute through the IPv6 routing infrastructure. We aim to see what can be learned about IPv6's deployment in each situation through the observation of live addresses. Despite the limited nature of the datasets, we see interesting variations between them.

* I would like to thank HEAnet and the IEDR for providing access to their logfiles.

This is not the only study that presents techniques to assess the state of the IPv6 Internet, but we believe this is the first to focus on addresses as the primary source of information. For example, [3, 11] analyse traffic seen at public 6to4 relays, considering indicators such as traffic levels, ports used and numbers of 6to4 clients. [1] describes a repository of traffic data, including IPv6 traffic, but also aims to anonymise the traffic, in the process scrambling much of the data that we aim to analyse. Work such as [8] focuses on routing tables and allocation of address blocks, but this exposes no information beyond the BGP prefix. Others have used active probing to measure IPv6 topology [4] or compare performance to IPv4 [2], focusing on the connectivity graph or performance of the network, rather than configuration details. Operators have also reported breakdowns of traffic volumes by application, using traditional indicators such as ports.

2 Address Analysis Technique

2.1 Prefix analysis

The breakdown of IPv6 address space is described by several IANA registries. The overall breakdown of address space is described in the `ipv6-address-space` registry. Smaller chunks are then described in more detail by other registries. In the case of other global addresses, we can use the `ipv6-unicast-address-assignments` to identify the RIR that addresses have been assigned to.

This analysis is the same sort of classification that can be performed on IPv4 addresses. However, in some cases an IPv6 address will provide details about how IPv6 has been deployed. In particular, we can identify users of 6to4 (`2002::/16`), Teredo (`2001::/32`, formerly `3ffe:831f::/32`) and 6bone allocations (`3ffe::/16`, formerly `5f00::/8`).

2.2 Host ID analysis

Just as the prefix can tell us where an address may be (al)located or if certain transition techniques are in use, the host-id can also give us information about how IPv6 is configured on a node. This sort of information is unavailable in IPv4 or in IPv6 if studies are solely based on address block or routing table information.

Perhaps the most common mechanisms for assigning host IDs are manual configuration (on routers and some servers) and autoconfiguration (based on the MAC address of a device). Autoconfiguration can usually be identified because of the way that MAC address are converted to host IDs. In particular, the dominance of EUI-64 based addressing and Ethernet/WiFi cards with vendor-assigned addresses leads us to expect `fffe` as the middle 16 bits of the host ID and the 7th bit of the host ID will be set.

ISATAP [12] is a technique that uses IPv4 as a layer 2 for IPv6, and it has a technique for generating an IPv6 host ID from the underlying IPv4 address. The construction leads us to expect the first quad (16 bits) of the host ID to be `0000` or `0200` and the second quad to be `5efe`, and these can be used as a test for ISATAP addresses.

ISATAP is not the only scheme that uses IPv4 addresses as a way to generate host IDs. IPv6 address parsers will usually allow the last two hex quads of the host ID to be

written as a traditional IPv4 dotted decimal. We use the following heuristic to identify host IDs that have the last two quads generated from a IPv4 address: an address is v4-based if (a) it is a 6to4 address and the second quad is the same as the seventh quad; or (b) the fifth and sixth quads are zero, the seventh and eighth quads are different, and the resulting IPv4 address would not be in IANA reserved/multicast address space.

We have based this heuristic on several factors. One is that 6over4 uses an IPv4 address padded with zeros and, as noted in [11], some 6to4 implementations do the same. Also, some sites derive their manually configured IPv6 address scheme from their IPv4 scheme. This test has weaknesses; we will discuss its effectiveness in Section 4.

Teredo also uses a special host ID based on two IPv4 addresses (the address of the NAT box and the address of the Teredo server). Since Teredo uses an easily identifiable prefix, we identify such host IDs based on the prefix.

In IPv4 networks, it is common for addresses to be assigned dynamically by DHCP from a pool. This currently seems less common in IPv6 networks, maybe because of the wide availability of autoconfiguration and relatively slow development of DHCPv6-capable servers. Concerns were raised because a fixed host ID, generated from a MAC address, would allow the tracking of devices as they moved from network to network. In response to this, a technique for randomly generating IPv6 host IDs was specified [10], which is now available in many IPv6 implementations. This *privacy addressing* uses a cryptographic hash to generate the host ID, and then clears the 6th bit.

```
00ad 00ba 00be 00d0 00da 00ed 0ace 0ada 0add 0ade 0b00 0b0a 0b0b 0baa 0bad 0bea 0bed 0bee 0c00 0c0b 0c0d 0cab 0d0b
0d0c 0d0d 0d0e 0dab 0dad 0deb 0dee 0ebb 0f00 0f0b 0f0d 0f0e 0fad 0fae 0fed 0fee abba b00b b0b0 b0de baba babe bade
baff bead beef c0c0 c0ca c0d0 c0da c0de c0ed c0ff cafe cede d00b d0d0 d0de dada dead deaf deed f00d f0ad face fade
faff feed 1337 0000 1111 2222 3333 4444 5555 6666 7777 8888 9999 aaaa bbbb cccc dddd eeee ffff 00ff abab
```

Fig. 1. Hex words users might use in IPv6 addresses.

This provides us with a technique to identify privacy addresses. A cryptographic hash should produce 0 and 1 bits in equal proportions. For a 63-bit output the Law of Large Numbers says that the majority of privacy addresses will have around 32 bits set. The actual technique used to identify privacy addresses is to first determine if the address can be identified as some other sort of address, and if not it is considered as a candidate privacy address. The address must then satisfy the following: the host ID must have the 6th bit clear; the host ID must have between 27 and 35 set bits; the first 32 bits must have between 9 and 21 set bits; the last 32 bits must have between 10 and 22 set bits; the host ID must not have two or more 'words' in it (as shown in Fig. 1).

These criteria are designed to cover the majority of privacy addresses, while rejecting patterns that are likely to have been manually configured, such as `::ffff:ffff` and `feed:babe:dead:beef`. We can calculate the proportion of random addresses that satisfy these conditions on the number of set bits as

$$\frac{1}{2^{63}} \sum_{\substack{9 \le i \le 21, 10 \le j \le 22 \\ 27 \le i+j \le 35}} \binom{31}{i} \binom{32}{j} \approx 0.7335. \tag{1}$$

Correcting for privacy addresses that are identified as being in some other type results in a insignificant change in this fraction. Thus, we expect this test to identify about three quarters of all privacy addresses.

The main type of host ID that we have not considered is manually-assigned host IDs. We cannot hope to identify all host IDs that are assigned directly by humans (or their scripts). However, humans are likely to opt for simple addresses that are easy to remember. One class of these are addresses ending in something simple, such as `::1` or `::53`. We attempt to identify these as addresses with the first 56 bits being 0, and call them *low* addresses. A class of address that humans are likely to be drawn to is those with regular patterns or words. Host IDs composed of 4 quads from Fig. 1 are *wordy*.

When attempting to identify a host ID, we take the first matching test from the following list: ISATAP, Teredo, autoconf, low, IPv4-based, wordy, privacy.

3 Data Sets

3.1 HEAnet Mirror Server

This dataset is based on the log files from the Apache server running on `ftp.heanet.ie`, a mirror server located in HEAnet, Ireland's research and education network. HEAnet's mirror server began offering IPv6 services publicly around May 2002, when a AAAA DNS record for the server was added. It mirrors a large number of projects, including Sourceforge, various Linux distributions, Apache, PuTTY, Mozilla, etc. It has a large user population and is the twentieth most visited site hosted in Ireland, according to Netcraft. There is no specific IPv6-related content on the site, though the software available may attract technically curious users.

Load on mirror servers can be highly variable. Peaks can be caused by new software releases or changes in available mirrors. For example, for a period `ftp.heanet.ie` was the only continuously operational Sourceforge mirror, resulting in increased load because of the sticky cookie used to select a Sourceforge mirror.

The dataset begins on 7 December 2003 and ends on 3 August 2007, over 1300 days. On some days during the period no data is available because of maintenance, service interruptions or log files no longer being available. One substantial gap is from mid-August 2005 to the end of 2005, due to an absence of log files.

From the beginning of the data, we have a list of the time and address of each IPv6 request to the server and summary IPv4 statistics. From 1 February 2005 onwards, Apache logs in the combined log file format entries for both IPv4 and IPv6 accesses are available. Fig. 2(a) shows the number of IPv6 hits (i.e. individual HTTP requests) on all days on which there were more than 1000 IPv6 requests. Daily IPv4 statistics are also shown where available, in some places interpolated from monthly statistics.

We aim to present statistics that account for, or make apparent, missing data and fluctuations in load. For example, to account for trends in IPv6 usage, we should factor out missing data or changes in load. One way to do this is to normalise by the IPv4 hits. For this to be valid, we need to know if IPv4 and IPv6 hits are correlated.

Fig. 2(b) shows a scatter plot of IPv4 vs IPv6 per-day hits. We plot points only where we have per-day statistics for both IPv4 and IPv6. The region excludes about 5

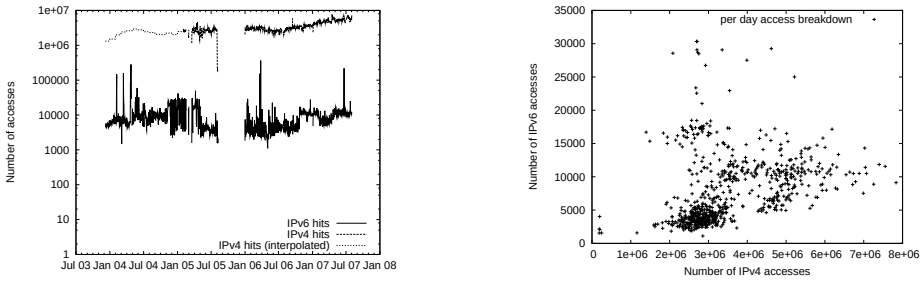

Fig. 2. HEAnet load statistics: (a) Per day hits (log scale), (b) scatter plot of IPv4 vs IPv6 hits.

outlying points. It seems the majority of days have the IPv4 load between about 400 and 1200 times the IPv6 load. However, there are a considerable number of points with higher IPv6 load. This suggests one should be cautious about blindly normalising by IPv4 hits, though IPv4 and IPv6 load do seem correlated.

We will be interested in the number of distinct addresses seen, as addresses are the fundamental unit of our analysis. Fig. 3 shows the number of distinct IPv6 addresses seen during each month. The number of distinct IPv4 addresses is also shown for comparison. We see that, except for the first few months, the pattern of fluctuations is similar for both IPv4 and IPv6, suggesting common causes for the fluctuations, such as those mentioned above. The dips in August 2005/2007 are caused by partial data for these months. Fig. 3 also shows the mean number of hits made per IP address. While these statistics were initially quite different, it seems as if they may be coming closer.

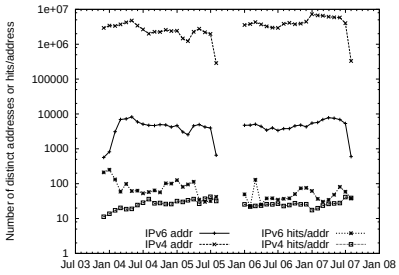

Fig. 3. Distinct addresses/hits per IP each month.

3.2 IE IPv6-Enabled Nameserver

The IE top-level domain is served by a number of IPv6-capable name servers. The IEDR, Ireland's domain registry, operate one of these, which has been advertised in the

root zone since September 2004. Log files showing all queries to this name server were available for the dates 22 April 2007–20 May 2007. The log files record the date of the DNS requests and the IPv6 address making the request. The server only deals with IPv6 requests, so no comparable IPv4 statistics are available.

3.3 Traceroute Data

In this section we consider quite a different dataset. The global IPv6 routing table is still quite compact, with only around 1000 prefixes present. We can consider what sort of IPv6 addressing is used to provide the routing infrastructure for this network. We consider tracrouting to the `::1` address of each prefix and recording the addresses revealed by the traceroutes. The aim is to reveal the addressing used to route between prefixes, without probing too deeply the internal structure of any prefix. Such a list of addresses should be dominated by addresses assigned to routers.

The list of addresses was collected in September 2007. A target list of 866 prefixes was prepared based on the IPv6 BGP table at HEAnet. Three different source addresses were used: one 6to4 address, one in a commercial ISP's PA space and one in HEAnet's PA space. We expect to see slightly different lists of addresses for each source address, because of both variability in routes and source address selection. For each source address, a list of intermediate router addresses was produced using traceroute. The three different source addresses produced 1558, 1687 and 1698 addresses respectively.

4 Results

4.1 HEAnet Mirror Server

We analyse the data from Section 3.1 first. Fig. 4(a) shows the proportion of IPv6 addresses in the 6bone, global production, 6to4 and Teredo address ranges from month to month. We plot the number of addresses falling into each prefix each month divided by the total number of distinct addresses seen during that month. We do not show results for a small number of local addresses, such as the loopback and link-local addresses.

We see substantial activity in the global and 6to4 address space, with the fraction of global production addresses showing an increasing trend. As expected, 6bone addresses were on a gradual decline, until a sharp drop in May 2006 before their retirement in June 2006. HEAnet did not carry routes to 6bone addresses after 6/6/2006, so after this date access to `ftp.heanet.ie` was not possible from 6bone.

Initially, we see a handful of Teredo-based addresses. However, since mid-2006 there has been a substantial increase in the use of Teredo clients. While this growth took place at the same time as early Windows Vista deployment, the User-Agent information indicates a mix of operating systems, mainly Windows XP, Linux and FreeBSD.

Fig. 4(b) shows the proportion of global addresses seen that were allocated by each RIR. Roughly, RIPE covers Europe, ARIN covers north America, APNIC covers the Asia/Pacific region, LACNIC covers Latin America and the Caribbean and AfriNIC covers Africa. As a mirror in Europe, we expect a majority of accesses to come from RIPE. Unsurprisingly, the statistics confirm this.

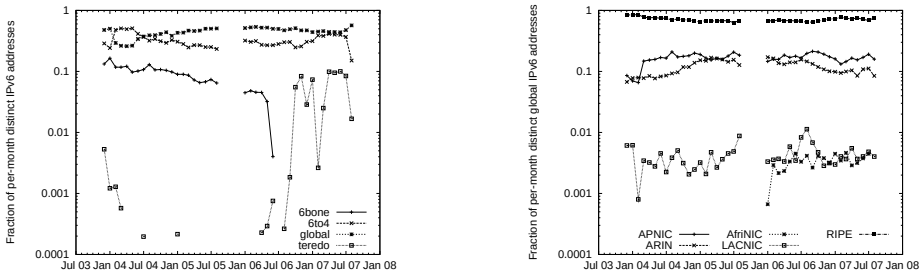

Fig. 4. Analysis of (a) all addresses by prefix, (b) global addresses by region.

Accesses from outside Europe are more interesting. These users have no particular reason to select a mirror in Ireland and so may give some indication of relative levels of activity. Initially, activity from ARIN and APNIC are at a similar level. Accesses from APNIC regions jump sharply in March 2004 and then slowly-increase. Accesses from ARIN gradually increase over time, catching up on the APNIC around February 2005, and then slowly decline. We cannot be certain if this is a change in the overall IPv6 node population in these areas, or particular content causing differential activity between regions. We see activity from younger registries at a low but consistent level.

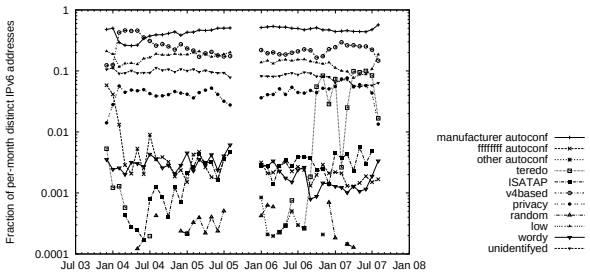

Fig. 5. Analysis of Host IDs.

Now we turn to the host ID. Fig. 5 shows how host IDs have evolved over time. Consistently there are about 10% of addresses that we cannot classify. The dominant technique is autoconfiguration based on vendor-assigned MAC addresses. The next most common technique seems to be IPv4-based addresses. Most (95%) of these addresses are 6to4 addresses using an IPv4-based host ID. Manually examining the remaining 5% of those identified as IPv4-based shows false positives, but the majority of results look correct. A substantial number of addresses allocated to BTexact's Tunnel Broker are identified as being IPv4-based: these also seem likely to be correct.

The next most common host IDs are addresses with only the low byte non-zero. These addresses do not seem to show evidence of any particular technology dominating, maybe indicating a mix of manual configuration and scripting. The wordy addresses are a smaller proportion of overall host IDs than the low addresses, though both types are substantially more common than would be expected at random.

About 4% of addresses are identified as privacy addresses. From Equation 1 we know our test under-reports, so the actual figure should be $4/.73$ or between 5.5%. Note, privacy addresses may, in a sense, be over-represented; while an autoconfigued address is fixed, a privacy address is periodically regenerated. We also show results for *random* addresses, which would have been classified as privacy addresses except the 6th bit was set. We see a tiny number of these *random* addresses. This indicates that the classification of address as as a privacy address is unlikely to include many false positives. When we inspect the random addresses, we see mostly random patterns and a few regular patterns that have been incorrectly matched.

Some addresses look autoconfigured but do not have the global bit set to a value we expected based on vendor-assigned MAC address. These may be generated from manually-assigned MAC addresses, may be soft MAC addresses on virtual machines, or may be manually-assigned host IDs. The data showed no addresses generated from the MAC range used by VRRP/CARP. We do find some addresses corresponding to autoconfiguration from the MAC broadcast address. This host ID may be the result of a failed Ethernet EPROM/faulty driver combined with autoconfiguration.

A small, but increasing, amount of ISATAP activity is visible. By comparing with Fig. 4(a) we see that by the end of our data the populations of 6to4, Teredo and ISATAP clients are roughly in a ratio of 30%:10%:0.3%.

4.2 IE IPv6-Enabled Nameserver

We now consider the data described in Section 3.2. Fig. 6 shows the results for the two months of data as log-scale bar charts. Activity is quite consistent over the two months. It is important to note that for a DNS query to be logged, there is no need for a TCP handshake to complete, and so there is no check for return routability. Thus, about 0.5% of queries come from 6bone prefixes and a handful of requests come from unassigned (`2000:1::1`), ULA ([6]) and documentation [9] addresses.

For comparison, Fig. 7 shows the corresponding results for `ftp.heanet.ie`. We expect a contrast between the client populations of these servers, because clients of mirror servers should not have much in common with recursive DNS servers. Indeed, when compared to `ftp.heanet.ie`, `ns6.iedr.ie` sees fewer requests from privacy, Teredo and ISATAP addresses. Even the proportion of 6to4 addresses is lower. The geographic distribution of the addresses also seems more even. We also see an increase in wordy and low addressing. This suggests that administrators shy away from transition mechanisms for recursive resolvers and opt for manually-assigned addresses.

4.3 Traceroute Data

Now consider the addresses observed in tracerouting across the IPv6 network from a commercial ISP, an NREN (HEAnet) and from a 6to4 network. Fig. 8 shows the

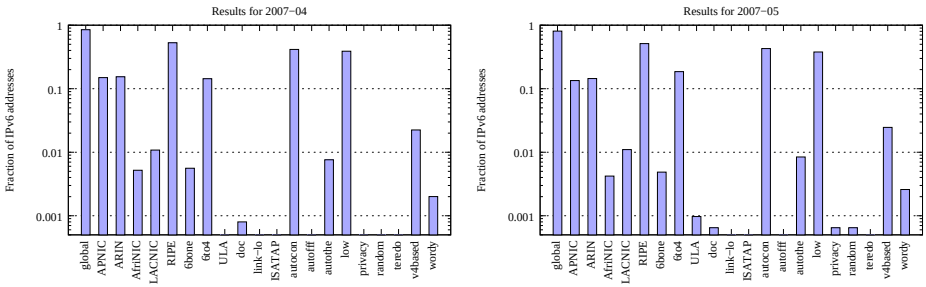

Fig. 6. Results for IEDR name server (log scale), April and May 2007.

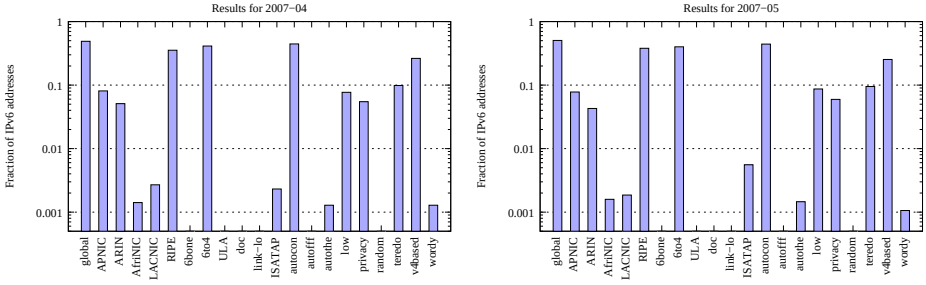

Fig. 7. Equivalent results for HEAnet ftp server (log scale), April and May 2007.

breakdown of addresses observed from each of these sources. Note that the results are quite consistent with each other, but show differences when compared to Fig. 6 and Fig. 7. In particular, almost all addresses seen are global IPv6 addresses and most host IDs are either low or IPv4-based. Some “autoconf” addresses are observed, however this is a misnomer in the case of routers, as routers can use EUI-64 based addressing, but do not assign their own addresses based on IPv6 autoconfiguration. Again, no addresses generated from the VRRP MAC address were observed.

There is an absence of Teredo and ISATAP addresses, and 6to4 addressing is uncommon except where the probes are sent from a 6to4 source address. If the probe is sent from a 6to4 address, source address selection should cause the router to choose a 6to4 address for the response, if it has one. When compared to the results in Fig. 6 and Fig. 7 we see a more even distribution across all 5 RIRs, representing the indiscriminately global nature of the traceroute. There is still some systematic bias in favour of RIPE as all the source nodes were located in Europe, but this dataset shows the most even geographic distribution of addresses. Otherwise, both the ftp and DNS server see a broader spread of address types than traceroute does.

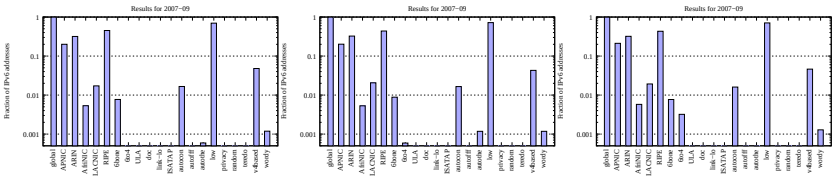

Fig. 8. Results for traceroute6 from Commercial ISP, HEAnet and 6to4, September 2007.

5 Conclusion

We have presented a technique for analysing aspects of IPv6 addresses. We applied this technique to three different datasets. The results produced have shown consistency from month to month, suggesting that the technique should be stable enough to identify trends in IPv6 deployment. From the traceroute data we also see that the results are consistent across different networks. Comparing results between datasets, the technique quantifies differences we expect based on the operation of IPv6 networks. These results build confidence that the technique produces meaningful results.

The results are consistent with what we expect of the IPv6 Internet, but we have not explicitly verified the accuracy of our technique. It would be interesting to do this by analysing data from known IPv6 networks. Accuracy might be improved accounting for the times addresses are observed and spatially/temporally adjacent addresses.

As further work, we would like to look at how subnets are being allocated within organisations and provide a more detailed study of host IDs, including tracing them and linking them with manufacturers. We would also like to explore applications of this technique to log analysis and adaption of service delivery/content.

References

1. K Cho, et al. *Traffic Data Repository at the WIDE project*, USENIX FREENIX Track, 2000.
2. K Cho, et al. *Identifying IPv6 Network Problems in the Dual-Stack World*, ACM SIGCOMM Workshop on Network Troubleshooting, 2004.
3. Y Kei and K Yamazakim. *Traffic Analysis and Worldwide Operation of Open 6to4 Relays for IPv6 Deployment*, Symposium on Applications and the Internet (SAINT), 2004.
4. CAIDA. *Macroscopic IPv6 topology measurements*. `http://www.caida.org/analysis/topology/macroscopic/IPv6`, 2006.
5. B Carpenter and K Moore. *Connection of IPv6 domains via IPv4 clouds*. RFC 3056, 2001.
6. R Hinden and B Haberman. *Unique local IPv6 unicast addresses*. RFC 4193, 2005.
7. C Huitema. *Teredo: Tunneling IPv6 over UDP through NAT.* RFC 4380, 2006.
8. G Huston. *IPv6 BGP/CIDR reports*. `http://bgp.potaroo.net/index-v6.html`.
9. G Huston, et al. *IPv6 address prefix reserved for documentation.* RFC 3849, 2004.
10. T Narten and R Draves. *Privacy extensions for stateless address autoconfiguration in IPv6.* RFC 3041, 2001.
11. P Savola. *Observations of IPv6 traffic on a 6to4 relay*. *ACM SIGCOMM CCR*, 35, 2005.
12. F Templin, et al. *Intra-site automatic tunnel addressing protocol (ISATAP)*. RFC 4214, 2005.