Where's that Phone?: Geolocating IP Addresses on 3G Networks

Mahesh Balakrishnan
maheshba@microsoft.com

Iqbal Mohomed
iqbal@microsoft.com

Venugopalan Ramasubramanian
rama@microsoft.com

Microsoft Research Silicon Valley
Mountain View, CA 94043

ABSTRACT

Cell phones connected to high-speed 3G networks constitute an increasingly important class of clients on the Internet. From the viewpoint of the servers they connect to, such devices are virtually indistinguishable from conventional end-hosts. In this study, we examine the IP addresses seen by Internet servers for cell phone clients and make two observations. First, individual cell phones can expose different IP addresses to servers within time spans of a few minutes, rendering IP-based user identification and blocking inadequate. Second, cell phone IP addresses do not embed geographical information at reasonable fidelity, reducing the effectiveness of commercial geolocation tools used by websites for fraud detection, server selection and content customization. In addition to these two observations, we show that application-level latencies between cell phones and Internet servers can differ greatly depending on the location of the cell phone, but do not vary much at a given location over short time spans; as a result, they provide fine-grained location information that IPs do not.

Categories and Subject Descriptors

C.2.5 [**Computer Systems Organization**]: Local and Wide-Area Networks—*Internet*; C.1.3 [**Computer Systems Organization**]: Other Architecture Styles—*Cellular Architecture*

General Terms

Measurement

1. INTRODUCTION

Smartphones connected to high-speed 3G networks are an increasingly important class of clients on the Internet. From the viewpoint of the websites they visit, such devices are virtually indistinguishable from conventional wired end-hosts, running fully functional browsers that display standard content with high fidelity. As with any wired host, a cell phone exposes limited information to Internet servers in the form of a User-Agent tag and an IP address.

In this paper, we examine the properties of the IP addresses exposed by cell phones to servers on the Internet; these IPs typically belong to application and network-level proxies within the carrier's network. Websites widely use IP addresses to identify end-hosts — for example, to prevent repeated voting on polls, or to prevent malicious activity. In addition, they often attempt to geolocate clients using IP addresses, using commercial services that map IPs to physical locations. Geolocation enables websites to implement more sophisticated functionality such as fraud detection, content customization and proximal server selection. IP-based identification and geolocation are known to work extremely well for wired end-hosts despite the prevalence of Network Address Translation (NAT) boxes and dynamic IP addresses [6].

Unfortunately, IP-based geolocation does not work well for cell phones. The graphic in Figure 1 shows the results of self-localization queries executed at seven different geolocation services by a cell phone in Mountain View, California. The query results from five of the services are not even localized to the same US state; later in this paper we'll see that the most accurate service shown does not work well for other locations. In this study, we show that the reasons for geolocation inaccuracy are two-fold. First, cell phone IPs are *ephemeral*, changing rapidly across HTTP requests

Figure 1: Answers to IP → Location queries provided by seven geolocation services; the actual cell phone is in Mountain View, CA.

IMC'09, November 4–6, 2009, Chicago, Illinois, USA.

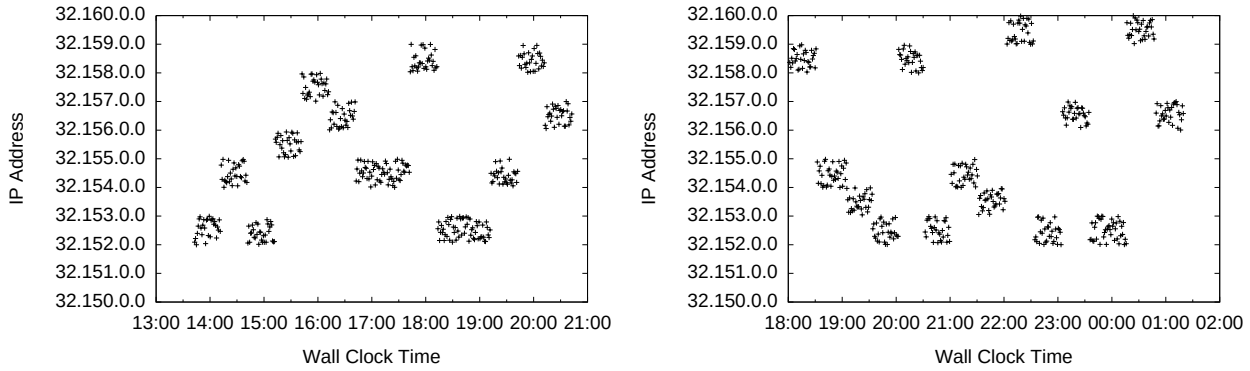

Figure 2: IPs sampled at 1-minute intervals on an HTC Touch Cruise (Left) and an Apple iPhone (Right) on the AT&T network, with radio resets every 30 minutes: all IPs were in the 32.152/13 range. The 16-bit prefix stays constant between resets and the 13-bit prefix across resets.

— as a result, each queried service in Figure 1 observes a different IP address for the same device, even though the queries are executed in quick succession within a span of five to ten minutes. Second, IP addresses for cell phones are *itinerant* — similar IPs can be exposed to a server by devices at geographically distant locations. In other words, IP addresses do not intrinsically embed fine-grained information on the location of mobile end-hosts. Consequently, IP-based geolocation is reduced to guesswork and different services produce vastly divergent answers for the same IP address.

To verify these two observations, we use two different sources of data. The first dataset is a collection of 1656 IP addresses obtained on the AT&T network by two different devices in Mountain View. We use this dataset to show cell phone IP addresses changing across requests spaced just 1 minute apart, albeit in constrained patterns and within specific ranges. The second dataset is a collection of 1299 AT&T IP addresses obtained from the logs of a service located in Redmond, Washington, belonging to devices primarily located in the Seattle area. We compare the Mountain View and Redmond datasets to illustrate that the range of observed cell phone addresses is nearly identical in two different geographic regions. Essentially, we show that IP addresses do not embed enough locality information to distinguish between these two regions.

In addition to making a strong case against IP-based identification and geolocation, we examine the possibility of using application-level latency measurements to geolocate cell phones. Our findings are that latencies on 3G networks are high compared to wired networks but exhibit low temporal variability at any given location. We show that certain cities that exhibit the same IP ranges on the AT&T network – such as Mountain View, Seattle and Albuquerque – can be distinguished from each other via application-level latency measurements.

The remainder of this paper is organized as follows: Section 2 shows that IPs are ephemeral, Section 3 shows that they are itinerant and Section 4 examines latency characteristics for 3G cell phones. Section 5 outlines related work and Section 6 concludes the paper.

2. HOW STICKY ARE CELL PHONE IP ADDRESSES?

To study the persistence of IP addresses on mobile devices, we ran an experiment on two different devices at Mountain View, CA on the AT&T 3G network: an HTC Touch Cruise (P3650) running Windows Mobile Professional 6.1 and an Apple 3G iPhone. On each device, we periodically visited a webpage at intervals of one minute and logged the IP address at the server. Every 30 minutes, we turned the radio on the device off, waited for 1 to 2 minutes, and turned it back on. We continued logging IPs in this fashion over a two day period. We would like to reiterate that these were IP addresses observed by the server on requests made by the devices; locally, the devices reported non-routable 10/8 prefix addresses that did not change and were not seen externally.

Figure 2 shows 7-hour traces from this experiment. The devices exposed a different IP address on every visit to the webpage, even though these visits were only separated by a minute each. Interestingly, within each 30 minute interval between radio resets, the IPs exhibited the same 16-bit prefix (for example, 32.154.x.x); however, when the radio was switched off and back on, the IPs moved to a different 16-bit prefix (for example, from 32.154.x.x to 32.158.x.x). Additionally, we only saw IPs in the range 32.152/13, indicating that in addition to the first 8 bits of the IP address (which are generic to AT&T properties), 5 more bits of the second octet are also static.

To determine whether our experience with transient IPs was specific to these two devices, we conducted a small study across multiple devices. We hosted a test website consisting of a form and two time-triggered auto-refreshes, designed to make a user interaction last for three to four minutes. The server logs IPs at four points during the interaction — once when the user first accesses the webpage, again when a button on the form is pressed, and twice more when the page subsequently auto-refreshes after timeouts that last 30 seconds and 120 seconds, respectively. Our purpose was to access this website from different devices and observe if their IP addresses changed within the very short time span on this interaction.

We tested 22 devices on this webpage, spanning 12 distinct device types — see Table 1 for the list of device types. Interestingly, 11 out of the 22 devices — and 8 of the 12

distinct device types — experienced IP changes during their interaction. The devices were geographically dispersed, with 9 devices in the Bay Area and the rest in Seattle (3 devices), Chicago (2), Atlanta (2), Albuquerque (1), Ithaca (4) and Germany (1); of these locations, we saw IP changes in the Bay Area, Seattle, Albuquerque and Ithaca.

Table 1 shows that we obtained variable results on iPhones. We had physical access to three iPhones in Mountain View, of which two exhibited the behavior shown in Figure 2, changing IPs every minute and changing prefixes across radio resets. The third did not change IPs between resets, but changed prefixes across resets. As Table 1 states, we ran 7 other iPhones through the short 4-minute website test, and two of them (located in Albuquerque and Seattle) changed IPs in this time span. Further experiments on these two iPhones showed that they exhibited the behavior shown in Figure 2.

Why do some iPhones change IP addresses rapidly while others do not? While we have not been able to conclusively answer this question, we did factor out three important variables. First, we eliminated the possibility of an anomalous cell tower by checking for IP changes from multiple cells in the Bay Area; in all cases, the iPhone continued to change IPs. Second, we swapped SIM cards between an IP-changing iPhone and the non-changing one; both iPhones retained their behavior, indicating that the phenomenon is specific to devices and not SIM cards. Third, we tested a single iPhone from multiple cities; the device continued to change IPs, ruling out region-specific behavior.

While most of the devices we tested were on the AT&T network, we did experiment with devices on four other major networks — Sprint (2 phones and 1 USB modem), Verizon (2 devices), TMobile US (1 device) and TMobile Germany (1 device). Of these, none of the Sprint devices changed IP, one of the Verizon devices changed IP and the sole TMobile US device changed IP. As a result, we do not believe IP address ephemerality to be exclusive to AT&T's network; however, we did not test on enough devices on any single network to determine ISP-specific behavior.

GSM-based networks typically use APN types to provide different IP addressing and connectivity requirements. Certain APN types enable static IPs and inbound connections, ruling out IP address changes. For APNs that provide dynamic IPs and do not allow inbound connections, it is possible that the frequency of IP address change is determined by APN parameters. More fundamentally, the need to prevent IP address space exhaustion is likely to push providers towards NAT and proxy mechanisms to handle large numbers of end devices. Network-level proxies can leverage ephemeral IPs to minimize long-lived client routing state, discarding each IP mapping as soon as the TCP connection is broken down. We believe that the IP address change seen in our study is evidence of such mechanisms within ISPs.

The implications of IP address ephemerality on mobile phones are extensive. In particular, many websites use IP-based blacklisting to restrict user access to content [6]. An ironic example that we encountered during this study is a popular commercial IP analytics website that uses IP-based blocking to enforce a limit on its demo interface, allowing users to lookup only a limited number of addresses per day. Using a cell phone, we were able to easily bypass this limit (although we must clarify that we only did this as a proof of concept and did not abuse the ability).

Device	Network	IP Changed?
iPhone	AT&T	Yes and No*
Samsung Blackjack	AT&T	Yes
HTC Touch Cruise	AT&T	Yes
Nokia E71	AT&T	Yes
BlackBerry 8310	AT&T	Yes
BlackBerry 8820	AT&T	No
HTC Tilt	AT&T	Yes
BlackBerry 8330	Verizon	Yes
Samsung SCH-i770	Verizon	No
HTC Touch Diamond	Sprint	No
Palm Centro	Sprint	No
BlackBerry 8900	TMobile	Yes
iPhone	TMobile**	No

*Of 10 tested iPhones, 4 changed IP and 6 did not.
**Located in Germany; all others were in the US.

Table 1: Devices tested for ephemeral IPs

3. GEOGRAPHICAL LOCALITY

An important use of IP addresses on the modern Internet is geolocation, where IPs are translated to physical locations. Geolocation is used by Internet services for a number of reasons, including server selection and content customization. We focus on the case where geolocation must be performed without the client's active participation. This is a requirement for uses such as fraud detection, where a site can raise a red flag if a user's profile lists a certain geographical address but her login IPs indicate different locations. Additionally, we assume that the Internet server does not have access to the ISP's information on cell phone locations — such information is usually not disclosed to third parties by ISPs.

A large number of websites routinely geolocate end-hosts under these constraints by using commercial geolocation services that maintain large databases with mappings from IP addresses to locations. These databases are generated through common sources such as *whois* databases and *traceroute* information, as well as proprietary analytics. For wired networks such as residential broadband hosts, the quality of geolocation is typically very good despite the presence of dynamic addresses and NAT boxes [6].

Clearly, ephemeral IPs do not pose a challenge for geolocation if they exhibit geographic locality — it does not matter to the geolocation service that a device obtains different IP addresses within a single session, as long as all these IPs are restricted to devices in the same geographical area. In this section, we aim to test if IPs obtained on the same network at different locations occupy different portions of the address space. We do so by comparing the Mountain View dataset described in the last section to a dataset of IPs collected on the same network (AT&T) at a different location.

We obtained this second dataset of IP addresses from the webserver of an internal service at Microsoft's Redmond campus. This service has 80 users who periodically log in and upload data — once a user connects to the service, the device sends data every 4 seconds. Almost all of the users of this service were located in the Seattle area, with a small fraction on other Microsoft campuses. The webserver had logs with IP addresses for a period of 51 days, spanning late March, April and early May of 2009. In total, the dataset

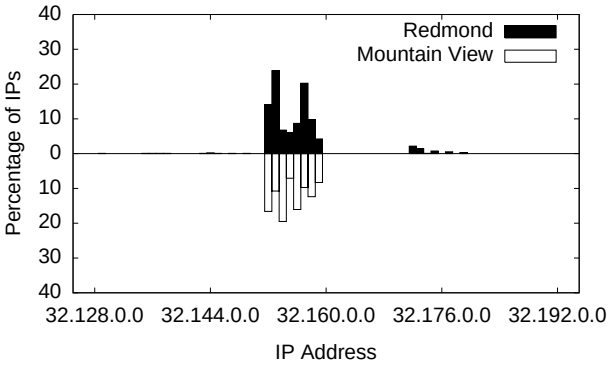

Figure 3: IP distributions on the AT&T network: 93.76% of the IPs in the Redmond (Top) dataset and 100% in the Mountain View (Bottom) dataset are in the 32.152/13 **range.**

contained 1526 total unique IPs.

Our first observation was that IP addresses in this Redmond dataset rarely changed within sessions; we attribute this to the 4 second interval between requests within each session, which is too short a time frame for the IP to change. The breakdown of devices in the dataset is heavily skewed towards AT&T; around 1311 of the 1526 IPs (or 85.91%) are on the AT&T network, out of which 1299 IPs are in the 32.x.x.x range, and 12 IPs are in the [209.183.32.0, 209.183.63.255] range. For the rest of our discussion, we focus on the 1299 AT&T IPs in the 32.x.x.x range, which constitute 85.12% of all the recorded IP addresses in the dataset.

Figure 3 shows a comparison of IP address distributions for this subset of the Redmond dataset with the Mountain View dataset. All the addresses in the Mountain View dataset lie within the 32.152/13 range (i.e., between 32.152.0.0 and 32.159.255.255). In the Redmond dataset, 1218 addresses (93.76%) are in this range. Of the remainder:
· 69 / 1299 (5.31%) are in [32.168.0.0, 32.184.255.255]
· 6 / 1299 (0.46%) are in [32.144.0.0, 32.151.255.255]
· 6 / 1299 (0.46%) are in [32.128.0.0, 32.143.255.255]

As we can see from Figure 3, the ranges occupied by the Mountain View and Redmond datasets are nearly identical, showing conclusively that the 16-bit prefix of the IP address does not embed fine-grained locality information: i.e., an Internet server cannot determine whether a cell phone is in Seattle or Mountain View on the basis of the prefix alone. We saw further corroboration of this conclusion in the 20-device study mentioned in Section 2 — the device in Albuquerque had a 32.159 prefix, similar to Mountain View and Seattle, while we saw the 32.136 prefix both in Chicago and Ithaca. Further experimentation with the device in Albuquerque showed that it obtained addresses in the same range as Mountain View and Seattle. Of course, it is still possible to locate phones at very coarse granularity based on the prefix — to determine if the user is closer to the east coast or the west, for instance.

3.1 Geolocation Accuracy in Practice

If our conclusions are correct, IP-based geolocation services should be unable to locate cell phones accurately. Figure 1 illustrated that different services give very different answers for a single device. To better quantify the accuracy of such services, we decided to run geolocation queries against a single service for the 1656 IPs in the Mountain View dataset, as well as the 1218 IPs in the AT&T portion of the Redmond dataset within the 32.152/13 range (we will discuss the results for the other IPs in this dataset separately). We chose the service that consistently gave us the best responses for the ad-hoc queries shown in Figure 1.

Figure 4 shows how the service performs on the two datasets — for each IP address in either dataset, we use the service to obtain a latitude and longitude, and then compute the distance between this coordinate and the actual location of the dataset (Mountain View or Redmond). At first glance, the service seems to be surprisingly accurate for the Mountain View dataset — nearly 65% of the addresses geolocate to within 200 miles of Mountain View. However, it performs extremely poorly for the Seattle dataset. Closer examination reveals that the service is returning an almost identical distribution of coordinates for both datasets, with a heavy incidence of Californian locations; the two large spikes in each graph correspond to the major urban centers in California.

Table 2 shows the distribution of locations returned by the service across states, for both datasets. The entry for "None" indicates responses where the service was unable to determine the exact location of the IP within the US, and instead returned blank values for the city/state fields and a generic coordinate in the center of the US. The answers for both Mountain View and Redmond are distributed over exactly six states on and near the west coast.

When we run the same experiment on the 81 other IPs from the Redmond dataset in the 32/8 IP range but outside the 32.152/13 range, the service no longer restricts its answers to the six states shown in Table 2. Instead, the distribution of answers has 71.6% in California, 16% with no state-level information, 4.9% in New York, and one IP each in Alaska, Indiana, Michigan, Ohio and Georgia.

Importantly, these results do not necessarily reflect the accuracy of the underlying techniques used by geolocation services, such as the use of network-level *traceroute* probes [9]. We have only shown that passive IP-based lookups that rely on the stationarity of existing IP to location mappings are not accurate for 3G phones. It is entirely possible that the mappings returned by the geolocation service were accurate at the time of measurement. If this is indeed the case, it is intriguing to note that the distribution in Figure 4 would be representative of the actual distribution of users on the west coast. Interestingly, an updated distribution could be obtained at any time by simply executing a series of geolocation queries for a single phone at long enough intervals that the IP address changes on each attempt.

4. LATENCY-BASED GEOLOCATION

Thus far, we showed that the IP address exposed by a cell phone over a 3G network does not embed locality information at reasonable granularity. Now, we examine the applicability of other techniques in the positioning literature that do not depend on the IP address. A number of these techniques assume the active participation of the end-host being located, running client software on the end-host; we rule these out since we focused on the setting of an Internet server attempting to locate a conventional cell phone accessing content via a browser. In any case, most cell phones are

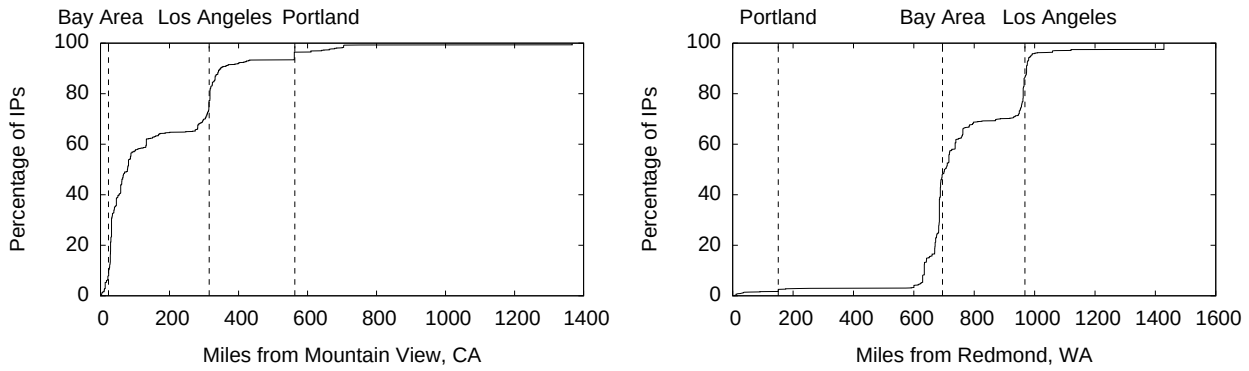

Figure 4: Accuracy of a geolocation service: Cumulative distribution of error between actual location of IP and the reported location, for the Mountain View dataset (L) and the Redmond dataset (R).

State	Mountain View Dataset	Redmond Dataset
CA	92.51%	93.10%
OR	3.14%	0.99%
WA	2.11%	2.05%
NV	0.84%	0.74%
None	0.66%	2.46%
UT	0.48%	0.24%
AZ	0.24%	0.41%

Table 2: Distribution of answers given by a geolocation service for the Mountain View dataset and the 32.152/13 range of the Redmond dataset.

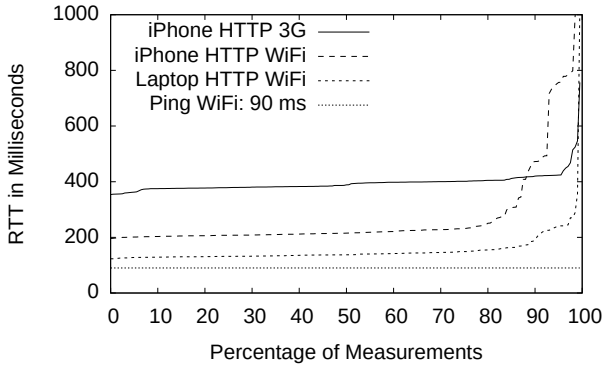

Figure 5: RTT measurements between Mountain View and Toronto.

equipped with GPS functionality, providing a trivial option if the cell phone can in fact participate actively.

Of particular interest are network positioning schemes that use latency measurements to locate end-hosts. While network positioning does not directly provide a physical coordinate for the end-host, it can provide a model for network latencies between the end-host and internet servers, which is sufficient for certain uses such as server selection. Also, geolocation of end-hosts can be achieved by combining network positioning information with the geographic locations of a few well-known nodes.

However, cell phones pose challenges for latency-based network positioning. In particular, the IP address exposed by the cell phone cannot be pinged, ruling out network-level measurements from the server to the cell phone. As a result, the server is forced to use application-level measurements; for example, by asking the browser to request an image in the HTML page and measuring the round-trip time (RTT) before the request for the image arrives back at the server.

To test if the application-level latency to an Internet server could be used to provide any information about the location of the cell phone, we first measured the latency from an iPhone in Mountain View to a webserver located on the east coast. The ping round-trip time between these two locations on a wired network was 90 milliseconds; we measured this by pinging the server from a laptop connected to a WiFi access point that led into a residential broadband network. On the iPhone, we measured two different sets of latencies: the latency to the server via the 3G network, and the latency to the server via the WiFi access point. In both cases, the RTT was measured and logged at the server through the HTML trick described. We measured the latencies by loading a webpage with the embedded graphic at 1-minute intervals.

Figure 5 shows these measurements and makes two important points. First, RTTs on the iPhone's 3G end-to-end path are very high at an average of 395 milliseconds, around 200 milliseconds higher than RTTs on the iPhone's WiFi end-to-end path. Second, RTTs on the iPhone's 3G end-to-end path exhibit low variability compared to the iPhone's WiFi path: 90% of the 3G measurements are within 16% (63 milliseconds) of the average. In fact, they also vary less than RTTs on the laptop's WiFi path; this indicates the variance might due to the WiFi/residential network, rather than differences in the iPhone's 3G and WiFi network stacks.

To further determine if application-level latencies could be used to pinpoint the location of clients, we obtained 200 RTT measurements each for iPhones on the 3G network from different cities: Mountain View, Seattle and Albuquerque. Figure 6 shows the results of this experiment. The RTT distribution for each city is very narrow and isolated from the others. We could observe no correlation within each distribution between an RTT and the IP address occupied by the phone for the corresponding request — all prefixes from 32.152 to 32.159 had equivalent RTT distributions.

However, Figure 6 comes with a major caveat — we observed very different RTT distributions for Seattle on different days. All the measurements in the figure were taken

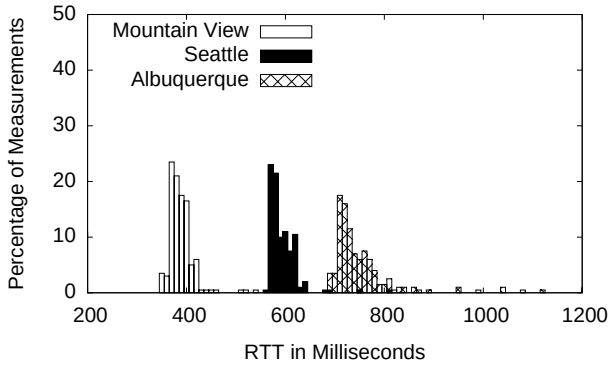

Figure 6: Distribution of Application-Level RTTs between a server in Toronto and iPhones in Mountain View, Seattle and Albuquerque.

in a contiguous 12-hour period spanning May 10th and May 11th. In measurements earlier that week, we observed vastly different RTTs for Seattle: a distribution centered around an average of 340 milliseconds, nearly 250 milliseconds lower than the average of the distribution for Seattle shown in Figure 6. In contrast, we did not measure any latency shifts for Mountain View, despite taking measurements periodically for over a week. Importantly, the earlier distribution for Seattle was also very narrow and easily distinguishable from the Mountain View and Albuquerque distributions.

Consequently, while latencies exhibit extremely low variance over short time periods, initial results seem to indicate that they shift massively over longer time periods. Further investigation is required to pinpoint the extent of these fluctuations and their underlying reasons. However, our findings indicate that application-level latencies can be used to distinguish cities that share the same IP range.

Together, the phenomenon of itinerant IPs and the latency measurements in this section possibly indicate infrastructure sharing across a broad area; for example, the west coast might have its own set of network-level proxies to which connections are load-balanced, with a different set of proxies for the east coast. This notion is supported by the fact that we see much higher latencies to Toronto from Albuquerque and Seattle when compared to Mountain View, even though the former two are closer in geographic distance. Also, the variation in latencies for Seattle across days could possibly be explained by coarse-grained load-balancing across proxies in different regions.

5. RELATED WORK

We are not aware of any other study to measure the stationarity of IP addresses of smartphones. For end-hosts on the traditional Internet, Casado et al. study the stationarity and opacity of client IP addresses in [6]. Their study found that IP addresses remained stationary over a long period of time (upto two weeks for 72% of the end hosts) despite DHCP reallocation and that networks behind NATs are typically small; mostly they consisted of two hosts, and in 99% of the instances had ten hosts or less. In a different study [13], Xie et al. found that more than 65% of dynamic addresses remain unchanged for at least a day.

This stationarity of Internet IP addresses has led to a host of successful techniques for geolocating Internet end-hosts without the use of GPS devices. Several commercial services [1, 2, 3] can locate an end host with reasonable accuracy, using publicly available databases that map domain names and IP addresses to geographical locations and other proprietary mechanisms. Several network-level techniques further enhance the accuracy of geolocation. IP2Geo [9] uses end-to-end latency and *traceroute* information to pinpoint geographic location. Octant [12] enables accurate geolocation by solving a system of geometric constraints.

The most effective technique known to geolocate a cell-phone is through the Global Positioning System (GPS) [7]. However, several techniques have been proposed to enable geolocation indoors and on cell phones not equipped with a GPS receiver. The Google Location Service uses the cell tower ID reported by the phone to locate it, conjunction with cell tower IDs reported by other, GPS-equipped phones [4].

More fine-grained geolocation can be obtained by tracking the strength of the radio signals received by the phone from different cell towers, as shown by Varshavsky et al. [11] for GSM-based cell phones. Their technique is inspired by other indoor localization systems built for different radio environments, namely ActiveBadge [8], Cricket [10], and RADAR [5] for Infrared, Ultrasonic, and WiFi radios respectively.

6. CONCLUSION

Modern cell phones are first-class clients on the Internet, providing functionality to users equivalent to wired end-hosts. In this paper, we studied the IP addresses exposed by cell phones on 3G networks to Internet servers. We showed that IPs can vary on short time scales on a single device, and that they contain very little information about the locality of the device; cell phones hundreds of miles apart share the same IP address space. These properties of cell phone IPs make IP-based user identification and geolocation almost impossible, hampering the ability of websites to blacklist users, display localized content, optimize performance and detect fraud. We also showed that application-level latencies between cell phones and Internet servers are high but exhibit low temporal variance, and can be used to distinguish between locations where the phones expose identical IP ranges.

Acknowledgments

We would like to thank our shepherd, Alberto Lopez Toledo, as well as the anonymous reviewers of this paper. Vinay Gupta, Kabita Mahapatra and Shailu Mishra at Microsoft provided us with traces and information on the state-of-the-art in cell phone geolocation. We would also like to thank the numerous people who volunteered their time and their phones for our experiments.

7. REFERENCES

[1] http://www.ip2location.com.
[2] http://www.maxmind.com.
[3] http://www.quova.com.
[4] http://googlemobile.blogspot.com/2008/06/google-enables-location-aware.html.

[5] P. V. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM 2000*, Tel-Aviv, Israel.

[6] M. Casado and M. J. Freedman. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *NSDI 2007: Fourth Usenix Symposium on Networked Systems Design and Implementation*, Cambridge, MA.

[7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag, 3 edition, 2001.

[8] A. Hopper, A. Harter, and T. Blackie. The active badge system. In *Proc of the INTERCHI Conference*, Amsterdam, The Netherlands, 1993.

[9] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *ACM SIGCOMM 2001*, San Diego, CA.

[10] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *MobiCom 2000: ACM International Conference on Mobile Computing and Networking*, Boston, MA.

[11] A. Varshavsky, E. de Lara, A. LaMarca, J. Hightower, and V. Otsason. Gsm indoor localization. *Pervasive and Mobile Computing Journal (PMC)*, 3(6):698–720, 2007.

[12] B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A comprehensive framework for the geolocalization of Internet hosts. In *NSDI 2007: Fourth Usenix Symposium on Networked Systems Design and Implementation*, Camdbridge, MA.

[13] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are ip addresses? In *ACM SIGCOMM 2007*, Kyoto, Japan.