Post-game Estimation of Game Client RTT and Hop Count Distributions

Grenville Armitage, Carl Javier, Sebastian Zander
Centre for Advanced Internet Architectures. Technical Report 060801A[1]
Swinburne University of Technology
Melbourne, Australia
{garmitage, cjavier, szander}@swin.edu.au

***Abstract*- In first person shooter (FPS) games the round trip time (RTT) (delay, or 'lag') between a client and server is an important criterion for players when deciding which server to join. Estimating the actual importance of this criterion can be challenging. Most game servers do not accurately log the RTT of either connected clients or potential clients (ones who only probed the server). Traffic traces also provide only IP addresses of hosts communicating with the game server. In this paper we propose a simple, active method of estimating the RTT between server and client when armed only with each client's IP address. For rough approximations this scheme works days or weeks after client IP addresses were collected. As jitter tends to be influenced by router hops we also discuss how to estimate the probable hop count between server and client. We illustrate our approach using data gathered from a Wolfenstein Enemy Territory server operating in Melbourne, Australia. This example shows our approach enabling after-the-fact comparisons between the RTT and hop-count distributions of clients who probe a server versus clients who actually join a server and play.**

Keywords- Game Traffic, Round Trip Time, Hop Count, Post-game estimation

I. INTRODUCTION

First Person Shooter (FPS) games are currently a very popular form of multiplayer networked game. Game clients probe game servers for information such as the current map, the number of current players on the server and the current network round trip time (RTT) between client and server. Potential players use this information to find suitable games and servers to join. The RTT (colloquially known as 'lag') between a client and server is of particular interest, as it strongly influences enjoyment in such fast-paced interactive games [1][2][3][4][5][6]. For this reason server operators and Internet service providers (ISPs) can find it useful to track and characterize the RTT tolerance of clients who frequent their servers. To do this they need a way to measure the typical RTT experienced by clients who probe and join, and those who probe and never join.

Collecting this information can be difficult. Any given FPS server may or may not have the ability to log RTT estimates for clients who join and play. Regardless, they are usually incapable of logging RTT estimates for clients who simply probe the server [7]. At best you might modify the server to log the IP addresses of probing clients. Even armed with an external packet sniffer program such as tcpdump you will be limited to collecting IP addresses rather than RTT estimates.

In this paper we propose an active method of estimating the RTT between a server and its clients when armed only with each client's IP address. For rough approximations this scheme works days or weeks after client IP addresses were collected. As jitter tends to be influenced by router hops we also discuss how to estimate the probable hop count towards each client IP address. Our proposal copes with clients going offline after they have played, IP addresses being reassigned to entirely different customers after being seen and logged by the game server, and network-layer filtering of ICMP traffic at (or near) the client end.

We demonstrate the use of this technique to estimate the distribution of RTT and hop count for game clients previously seen contacting a Wolfenstein Enemy Territory server based in Melbourne, Australia [8]. The derived results provide insights into the geographic and topological distributions of clients who chose to play and those who chose not to play on this particular server. As a side benefit, investigation of RTT versus hop count across the set of clients provides a perspective on Australia's overall 'distance' to hosts across the rest of the Internet.

The rest of the paper is organized as follows: section II describes the proposed measurement methodology. Section III demonstrates the use of this methodology on client IP addresses gathered from a specific game server. Section IV concludes the paper.

II. RTT AND HOP-COUNT ESTIMATION METHODOLOGY

Our scheme relies on actively testing, from the server's current location on the network, a subset of IP addresses known to represent clients who played or probed a particular game server. In this section we describe the key assumptions, the sampling technique, and the combination of two active probe techniques (ping and traceroute) used to estimate past RTT and hop count distributions.

A. *Assumptions*

A key assumption underlies this technique: any given

[1] A revised version of CAIA-TR-060223A, February 2006

A 4-page version of this paper was accepted at Netgames 2006 (Singapore, Oct.30th-31st 2006) and presented as a poster

IP address is presumed to be roughly the same distance away today (measured by RTT and hop count) as it was when first logged. We observe that most game clients connect via consumer ISPs whose end-user IP address ranges are unlikely to move around much topologically. Clearly the validity of this assumption degrades over time. However, it should be acceptable over weeks or months (in the absence of an ISP losing an entire IP address block to another ISP in a different country or region of the Internet).

B.*Sampling the Client IP Address Set*

An active game server is likely to see thousands if not millions of separate IP addresses over periods of months. To simplify the subsequent active probing process we select a subset of logged client IP addresses to represent the characteristics of the path between our server and all clients.

Our simplifying assumption is that clients whose IP addresses fall under a common CIDR prefix will share much the same path back from the server towards each client. For example, in consumer ISP contexts the hop count measured to one IP address in a /24 is likely to be the same as the hop count to any other IP address in that same /24. Furthermore, we suggest that RTT measured to one IP address in a /24 is likely to be representative of the RTT to any other IP address in that same /24. (It is true that consumer ISPs may use last-hop access links with quite different latency characteristics – such as dial-up, cable modem or ADSL. Nevertheless, it is not unreasonable to assume that IP addresses within a single CIDR prefix are served using a single access technology.)

In section III we illustrate this approach as follows: where multiple clients IP addresses share a common /24 prefix we randomly select only one of those client IP addresses to measure for RTT and hop count. (Longer or shorter prefix lengths may be utilized if it is known that IP addresses in certain ranges are allocated along particular prefix boundaries.)

C.*Clients need not remain reachable*

It is unlikely that a game client seen in our server logs will still be active on the Internet days, weeks or months after the fact. At the time we launch our RTT measurement the IP address may have been reassigned to someone entirely different or the client may be turned off.

We do not actually require the original client to be present at the logged IP address. It is sufficient that some entity responds to ICMP Echo Requests directed towards each selected IP address.

In practice it is possible for our ICMP Echo Requests to elicit no response from selected IP addresses. The target may simply be turned off or IP-layer filtering may be active along the path towards the target. In this case, we utilize traceroute to probe the path out towards the selected IP address and derive RTT and hop-count estimates.

In principle ICMP may be blocked anywhere along the path towards the targeted client IP address, skewing traceroute's results. However, by comparing the results from client addresses that responded to ping and those that needed traceroute we can estimate an adjustment to the traceroute-derived RTT and hop-count results.

D.*Measuring RTT and Hop Count*

Figure 1 shows the basic probe sequence for one IP address selected from the set of client IP addresses to be tested. If ping fails to establish an RTT estimate (for whatever reason), we approximate the RTT estimate by measuring the RTT (again using ping) to the last IP hop seen using traceroute. If traceroute's last reported IP hop cannot itself be pinged we use the RTT estimate provided by traceroute itself.

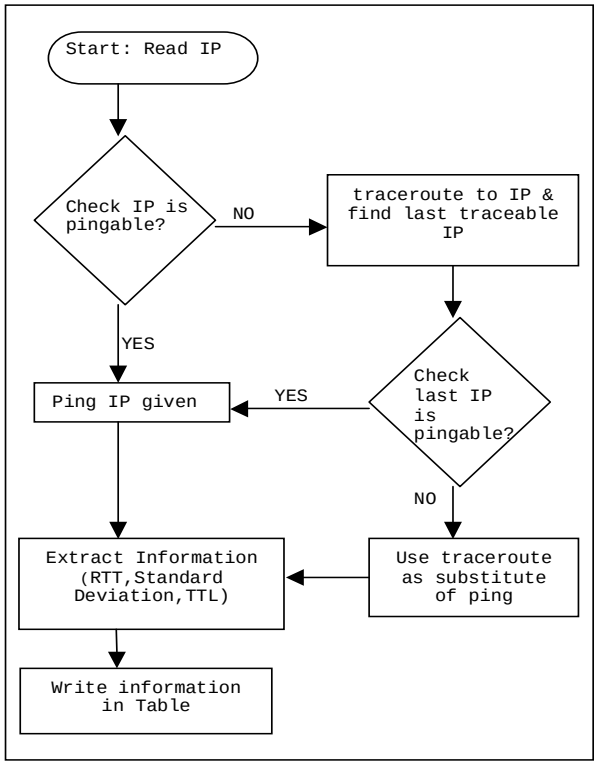

Figure 1: Algorithm for Estimating RTT to Previously Identified IP Addresses

Each selected IP address is pinged ten times at two-second intervals. The smallest of the ten ping results is chosen as the RTT estimate most likely to be unaffected by transient congestion along the path. The standard deviation is also calculated to provide some indication of how stable the path was during all ten RTT estimates. Spacing the pings every two seconds minimizes the chances of our efforts being misinterpreted as a denial of service attack on the target ISP.

Ping can fail for a number of reasons - the destination host no longer exists or is not switched on, the ICMP echo requests are blocked by the end user's home firewall or the ICMP echo requests are being blocked by ISP firewall policy somewhere along the path. If ping fails we follow up with traceroute. The last hop successfully reported by traceroute is pinged and the RTT recorded. If ping does not work, we record the RTT estimated by traceroute itself.

Hop count is estimated from the TTL field of ICMP messages being returned in response to ping or

traceroute. Since the TTL is decremented once per hop back towards our location, we can estimate the number of hops traversed, by subtracting the final TTL from the initial TTL. (Note that if traceroute is used from a Windows system the outbound and returned packets are both ICMP. When traceroute is used from a unix-like system the outbound packets will be UDP and the returned packets ICMP.)

Game clients are most likely found running on Windows hosts (and to a much lesser extent, Linux hosts). Such hosts typically utilize an initial TTL of 32, 64, 128 or 255 [9]. We believe most consumer routers are likely to respond to traceroutes from a similar possible set of initial TTLs. Since it is generally believed that few Internet hosts are more than 32 hops away from each other [9] we assume the initial TTL value of a packet as the smallest of 32, 64, 128 or 255 that is larger than the final TTL in each received ICMP packet.

Traceroute should be configured to probe no more than 32 hops away. This substantially reduces the time taken to estimate the last hop of an IP address that cannot be pinged directly (since traceroute must reach its maximum TTL before the identity of the last successfully reported hop can be confirmed).

Ultimately every selected client IP address ends up being associated with an RTT and hop-count value in one of four categories:

(A) pinged the client IP address directly
(B) pinged the last hop reported by traceroute
(C) used traceroute's RTT estimate to the last hop reported by traceroute
(D) RTT and hop-count estimated based on the last hop reported by traceroute (B and C collapsed into a single category)

E.*Adjusting the Last Hop Reported by Traceroute*

Two simple sanity checks should be applied to the last-hop returned by traceroute. If the reported last-hop comes from private address space (e.g. 192.168/16 [10]) or has a different country code than the target client IP address (as reported by a database such as GeoLite Country [11]) we do exclude this data point from further analysis.

Results from category D are then adjusted to estimate the RTT and hop-count to the client IP addresses that could not be pinged directly. First we plot the distribution of RTT and hop-count values returned in categories A and D on separate cumulative distribution curves. Over thousands of tested IP addresses in each category the distribution curves should look similar, but offset from each other. The median difference between the curves of both categories indicates the offset to be applied to RTT and hop-count results in category D.

F.*Limitations and Considerations*

Most of the assumptions listed earlier rely on observed operational traditions within consumer ISPs. Such traditions are not necessarily mandated by IETF standards or specifications, and may not be universally true over time. (For example, not using a common /24 prefix to cover multiple clients connected through different access technologies having diverse RTT characteristics.)

Another limitation is that our RTT measurements are not taken under the same network conditions that existed while each client was accessing the server. For example, ICMP packets do not have the same length distributions as game packets, leading to slightly different serialization delays along the path. It is also well known that routers do not handle ICMP packets quite the same way as regular UDP or TCP packets, potentially leading to slight over-estimation of RTT to the selected client IP addresses [13]. However, we suggest over-estimating by few milliseconds is tolerable in the context of game clients from around the planet exhibiting tens or hundreds of milliseconds RTT.

In addition, we assume peering agreements along the path to each client are essentially unchanged. In principle such agreements may change at any time, altering the internal topology of the Internet between access ISPs. Thus the RTT and hop-count distributions measured today may differ significantly from those experienced by individual clients when they actually played on your server.

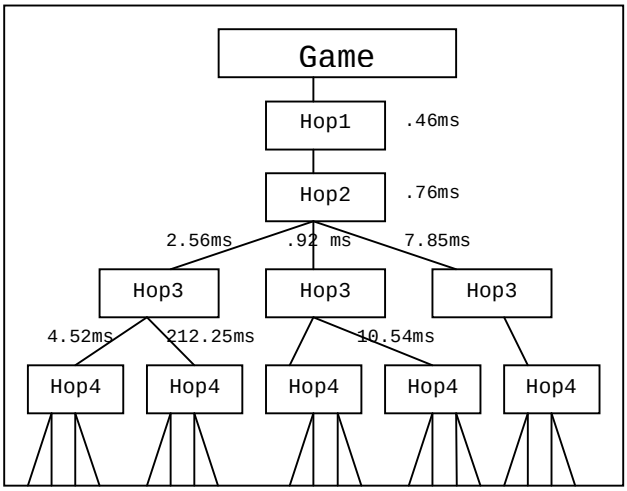

Figure 2: Hops Along Different Paths May Contribute Quite Different Latencies

It is also worth keeping in mind the variable relationship between RTT and hop-count (noted in previous related work, e.g. [14]). Along a given route RTT usually increases with increasing hop count. However, different routes may exhibit quite different relationships between RTT and hop count. Physically short hops will contribute far less propagation delay than physically long hops. The next hop towards one IP address may jump a few metres inside an ISP, yet the next hop to another IP address may involve thousands of kilometres between continents. For example, Figure 2 illustrates the diversity of paths and RTTs seen at 1, 2, 3 and 4 hops away from the server analysed in section III. Along the far left hand branch hop 3 is 2.65ms away. If we continued along the left branch hop 4 is 4.52ms away, but if we went right hop 4 would be 212.25ms away.

III. ILLUSTRATION USING A GAME SERVER BASED IN AUSTRALIA

In this section we illustrate the use of our RTT and hop-count estimation technique. Client IP addresses, gathered from an Australian-based Wolfenstein Enemy Territory (ET) server [12] are used to build plausible and useful insights into the distribution of players and non-players who visited the server.

A. *Background*

In 2005 we published an analysis of server-probe traffic impacting two ET servers based in Australia [8]. That research differentiated between clients who actually played on each server, and clients that were only ever seen probing each ET server (for updated game-state, RTT and map information). We showed that a modestly utilised FPS server is inundated with many hundreds of thousands of probe queries per week, regardless of how many people actually play on the server. Data was gathered over 20 weeks between November 2004 and March 2005 from servers based in the cities of Melbourne and Canberra. Probe and game-play traffic was analysed for its daily and weekly fluctuations by volume and approximate geographic origin (using Maxmind's GeoLite Country [11], which claims to be 97% accurate). Over the 20 week period probe traffic contributed roughly 16 million flows, 36 million packets and 8 gigabytes of data transfer in and out both the Melbourne and Canberra servers. By contrast, game-play accounted for roughly eight thousand flows, 755 million packets and 116 gigabytes of traffic in and out of the Melbourne server. (The Canberra server was less popular and saw far less game-play traffic.)

As a side effect we ended up with roughly 2.4 million distinct client IP addresses from the Melbourne server for which we had no RTT or hop-count information. Neither server had been modified to log its internal RTT estimates for clients who actually played, nor could they meaningfully estimate RTTs for clients who simply probed (without joining). Unfortunately, due to disk space limitations we had not kept tcpdump files that would have provided TTL information from which to estimate hop-counts.

Table 1: Subnet Reduction of IP Addresses

	Initial No. Of IP addresses	**Reduced No. Of IP addresses**
Game Flows	5,469	4,252
Probe Flows	2,397,879	325,707

Hoping to gain some insights into the differences between clients who played and probed, we decided to compare the RTT and hop-count distributions of each class of clients. We reduced the 2.4 million client IP addresses by selecting one IP address at random from groups sharing common /24 prefixes. Table 1 shows the significant benefit of this reduction – from 2.4 million we ended up with roughly 330,000 IP addresses to actively test. (The most significant reduction involved IP addresses who were seen to probe rather than play our server.)

B. *Performing the Active Measurements*

Active scans were performed from a FreeBSD 5.4 host (a 2.8GHz Intel Celeron with 1 GB of RAM) on the same IPv4 subnet as the Melbourne ET server used in [8]. To speed up measurements we ran fifty parallel instances of the algorithm explained in section II.D. Each instance tested a non-overlapping set of client IP addresses from the 'reduced' set in Table 1, and was launched at a random time relative to each other (to minimize correlated bursts of outbound ping or traceroute traffic). With fifty instances running the CPU load fluctuated between 3% and 5%, suggesting CPU load would have minimal impact on RTT estimates reported by ping or traceroute. Averaged over all IP addresses in category A and category D (section II.D) the ping/traceroute sequence took 1.45 minutes per address. (Adjusting the FreeBSD 5.4 kernel's default tick rate from 100Hz to 1000Hz was also necessary to provide 1ms resolution to ping RTT estimates. Versions since FreeBSD 5.4 now ship with a default tick rate of 1000Hz [16].)

C. *Summary of Raw Results*

Our raw results were post-processed to remove anomalous data points before creating the statistics shown in Table 2. 'Game flows' refers to the class of clients who established game-play traffic flows to the server, whereas 'probe flows' refers to the class of clients who established short-lived probe-only traffic flows to the server.

Table 2: Game Flow and Probe Flow Results

	Game Flows	**Probe Flows**
Number of IP Addresses	4252	325,707
Ping directly	28%	26%
Ping last hop from traceroute	63%	62%
Used traceroute for RTT computation	9%	12%

In brief, approximately:

- 2% of traceroute-derived data points were removed because the last hop IP address was not in the same country as the target client IP address.
- 0.004% of traceroute-based data points were eliminated because they returned a private IP address [10] as the last hop.
- 2.6% of game flow IP addresses and 1.4% probe flow IP addresses were removed because the RTT was calculated to be over 1000ms, or the standard deviation over ten RTT samples was over 100ms.

D. *Accuracy of RTT Estimations*

Figure 3 shows the distribution of both dataset's standard deviation. More than 90% of the RTT estimates have a standard deviation under 10ms, suggesting the estimation process was fairly consistent over the 10 pings.

Probe flows show a slightly higher standard deviation

because (as we discuss later) clients who only probed were typically 'further away' (at higher RTT and higher hop count) than game flow clients. Higher hop count means more router hops – and thus congestion points - at which jitter may potentially be introduced.

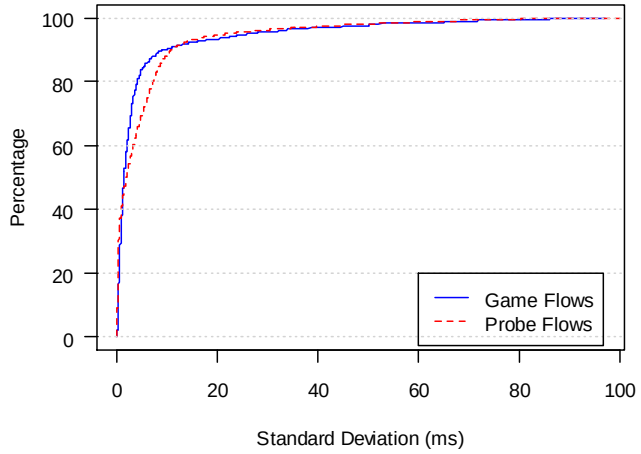

Figure 3: Probe and Game Flow – Standard Deviation of RTT Estimates (CDF)

E.*Validity of Using Traceroute to Determine the Last Hop*

One of our implicit assumptions is that traceroute can be used to identify an IP address topologically close to the target IP address when the target IP address itself does not respond to ping. Ideally, 'close' would mean we find the last hop before the target IP address. Our results suggest this assumption is reasonably valid.

From Table 2 we see that IP addresses associated with 28% of game flows and 26% of probe flows responded to a direct ping. We call these 'pingable' IP addresses. The rest are 'non-pingable', where we are approximating the desired data point by measuring RTT and hop count to the last hop successfully identified by traceroute.

Cumulative distribution function (CDF) plots for both pingable and non-pingable data points reveal that non-pingable clients seem to be one or two hops and 10-30ms closer than pingable clients. This suggests our traceroute technique is, in fact, generally identifying an IP device one or two hops from the target IP address.

Figure 4 and Figure 5 show the CDFs of measured hop counts for game flow and probe flow IP addresses respectively. If the non-pingable curve is moved right by one hop (game flows) or two hops (probe flows) the distributions for pingable and non-pingable flows are approximately identical. This is consistent with the non-pingable data points being derived from an IP entity one or two hops closer than pingable data points.

A similar, although slightly weaker, observation can be made based on RTT estimates. Figure 6 and Figure 7 show the CDFs of estimated RTT for game flow and probe flow IP addresses respectively. In this case we found the distributions for pingable and non-pingable flows are roughly the same if the non-pingable curve is shifted right by 20ms.

Consequently, for the rest of our analysis we adjusted all non-pingable data points up by 20ms and one or two hops (for game and probe flows respectively).

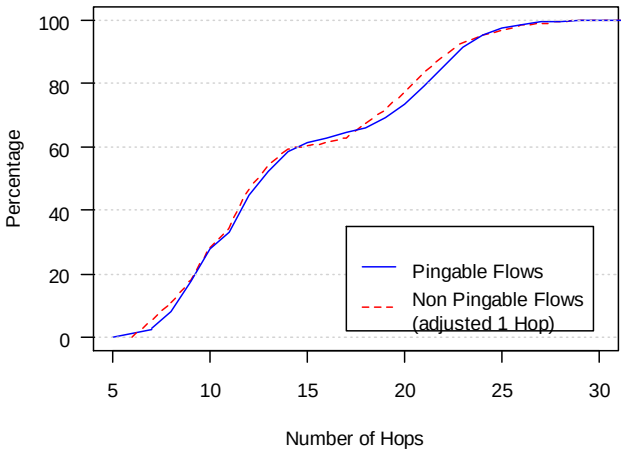

Figure 4: Game Flows – Pingable & Non Pingable Hop Count CDF

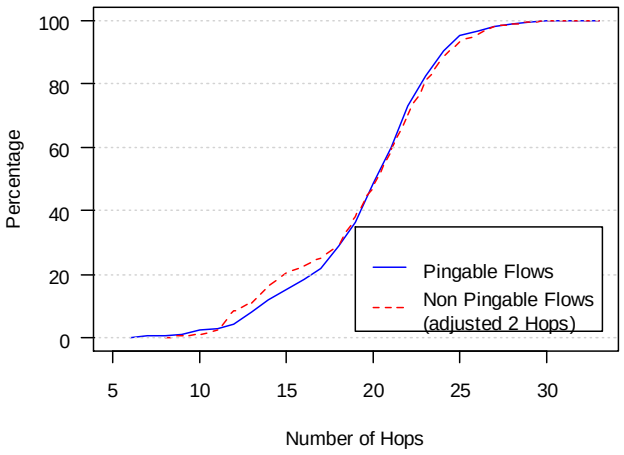

Figure 5: Probe Flows – Pingable & Non Pingable Hop Count CDF

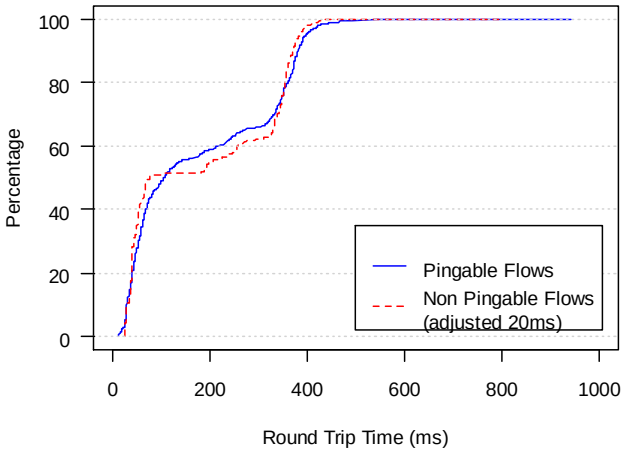

Figure 6: Game Flows – Pingable & Non Pingable Round Trip Time CDF

(These offsets are plausibly due to the common use of consumer-grade last-hop access technology such as dial-up, ADSL or cable modem. The actual game clients whose IP addresses were 'non-pingable' would have probably been 10ms to 30ms further away than the ISP router interface we were ultimately able to ping. We appear to be on relatively safe ground in treating the adjusted traceroute-derived data points as equivalent to pingable data points.)

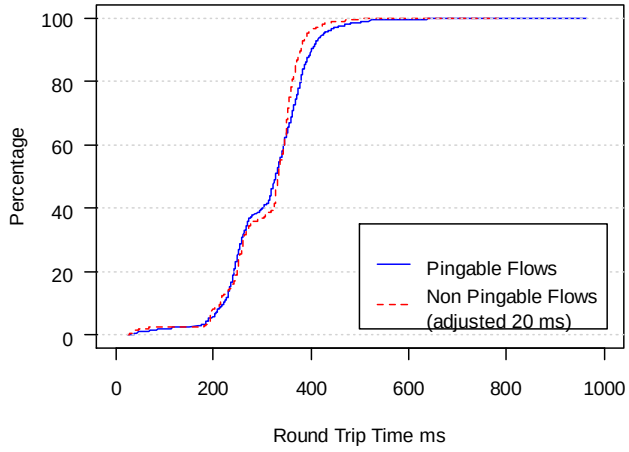

Figure 7: Probe Flows – Pingable & Non Pingable Round Trip Time CDF

F.*Geographical Distribution of Game Clients*

Using the GeoLite Country database [11] we identified IP addresses from 54 countries amongst game flows and 138 countries amongst probe flows. As previously reported in [8], a vast majority of game flows were attributable to only a small number of countries. Australian players accounted for 57% of the game flows, the next highest being Poland with approximately 8% of game flows, followed by USA and Germany with 4-5% each. By contrast, probe flow demographics were quite different European countries contributed to 52% of probe flows, with the USA contributing another 30% of probe flows.

Using the technique in section II allowed us to extend the results from [8] to reveal the topological consequences of being from different countries. Figure 8 shows the distribution of hop counts for both game flow and probe flow clients from a number of countries. Australian clients are 5 to 15 hops away while international clients are at least 10 hops away. (As implied by Figure 2, 10 to 15 hops to international clients are likely via quite different and physically longer paths compared to the Australian clients who are also between 10 and 15 hops away.)

Figure 9 shows the distribution of RTTs for clients from a number of countries, along with the average RTT from each of the countries. Australia has an average RTT of 56ms (with almost all clients being below 100ms) while clients from other countries have RTTs of at least 180-200ms.

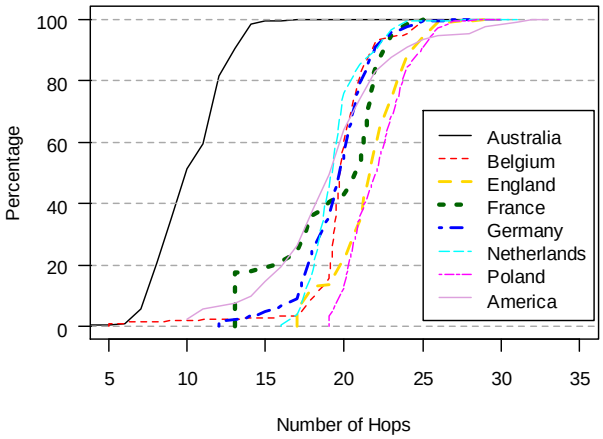

Figure 8: Hop Count per Country (CDF)

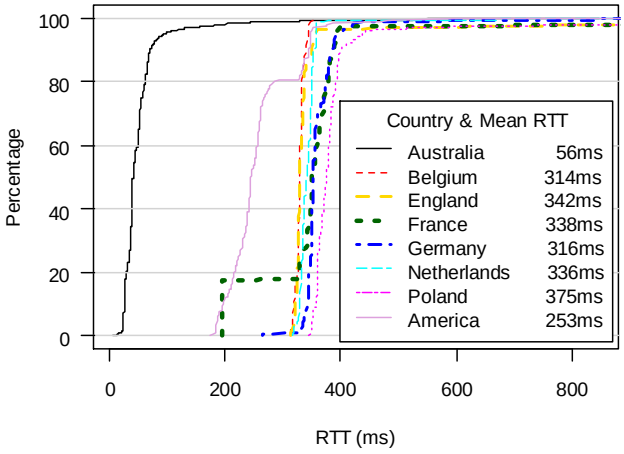

Figure 9: Round Trip Times per Country (CDF and mean)

G.*RTT and Hop Count Analysis*

Comparing the RTT distributions of game and probe flows (as shown in Figure 10) makes clear the correlation between RTT and people's decision to play or not play. Around 50% of game flows have RTT less than 100ms, and 60% of game flows have an RTT of less than 200ms. By contrast, the majority (over 90%) of probe flows (people who subsequently chose not to play on our server) originate from clients with RTT over 200ms. This provides indirect support for previously published work that puts FPS player tolerance for RTT between the high-100s and low-200s of milliseconds [1][2][3][4][5].

A similar comparison is provided by Figure 11, which compares the hop count distributions for game flow and probe flow clients. Less than 10% of probe flows appeared with hop count under 13, whereas 60% of game play flows occurred with hop count under 13.

Figure 11 also provides a clear indication that no game-playing clients were closer than 5 hops, and confirms the existence of two distinct communities of players – those between 5 and 15 hops away, and those between 17 and 25 hops away. On the other hand, the

community of probe-only clients is clustered strongly between 10 and 25 hops away from our server. Based on Figure 8 the majority of these probe-only clients (particularly over 15 hops away) reside outside Australia.

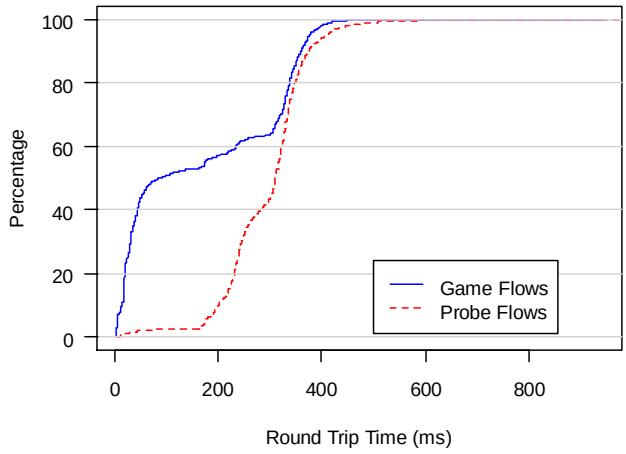

Figure 10: Probe & Game Flows – Round Trip Time CDF

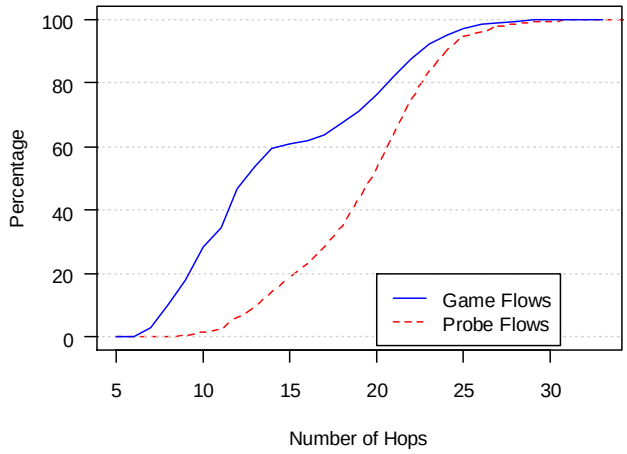

Figure 11: Probe and Game Flows – Hop Count CDF

Relationships between apparent geographic origin, RTT and hop count are shown in Figure 12 (for game flows) and Figure 13 (for probe flows). Both figures show graphs of average RTT versus hop count for flows originating in five different countries.

Both graphs clearly reveal that RTT experienced by players outside Australia is dominated by the paths taken just to get to and from Australia itself. We can see that most Australian clients are between 5 and 15 hops away, and less than 100ms. Most American clients are between 10 and 26 hops away, and between 180 and 300ms. Clients from France, Germany and Poland tend to be 16 to 25 hops and 320 to 400ms away.

For destinations outside Australia there is one or more long-haul international links before traffic distributes itself around within their home country. In-country RTT versus hop count has a fairly modest gradient in both graphs. This reflects the fact that while IP paths in-country cover small geographic areas they may have many hops through closely located ISP equipment racks or Internet exchange points. (A dip in the mean RTT versus hop count at a couple of places is a consequence of aggregating the RTTs from clients reached through diverse in-country paths, similar to what we noted in Figure 2.)

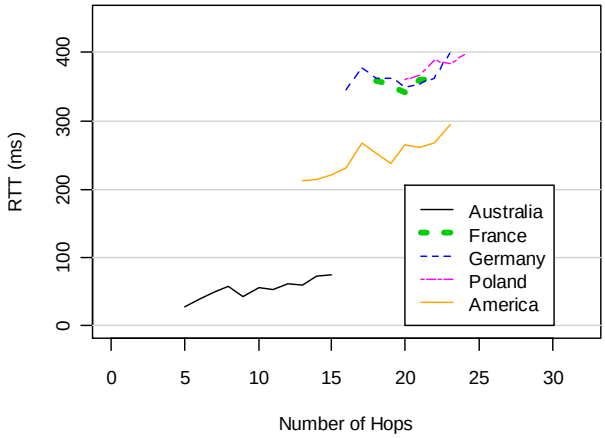

Figure 12: Game Flow – Mean Round Trip Time vs. Hops per Country

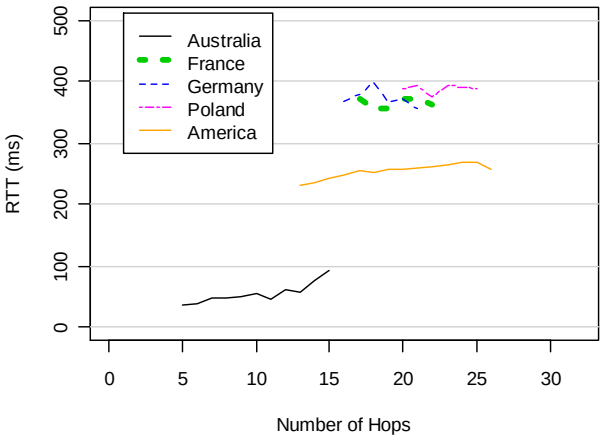

Figure 13: Probe Flow – Mean Round Trip Time vs. Hops per Country

IV.CONCLUSIONS

Although game servers can be instrumented to log RTT estimates of clients who actually play on a server, it is difficult to log the RTT experienced (or perceived) by clients who simply probe a server without playing. This paper describes a simple technique involving ping and traceroute to establish RTT and hop-count estimates, after the fact, between a game server and game clients who may no longer be attached to the Internet.

We assume that RTT and hop-count estimates must be derived at some point in time long after the client IP addresses were logged at a game server. We further assume that, by virtue of being seen playing (or probing) a game server, each client IP address is most likely associated with a consumer Internet connection. This enables a key simplification – the RTT and hop-count to every client IP address under a common CIDR prefix

will be approximately the same. So, for example, multiple client IP addresses from the same /24 can be measured by estimating the RTT and hop-count to just one of them chosen at random. This step can reduce millions of IP addresses down to thousands for subsequent active probing.

Since clients may come and go, and ping's ICMP echo request/reply packets are often blocked by personal firewalls near the target host, it may be necessary to use traceroute to identify an IP router close to a target client IP address. We show how the distance (in RTT and hops) between a traceroute-derived last hop and the actual target client IP address may be indirectly inferred when large numbers of client IP addresses are available.

Our approach is demonstrated using client IP address data collected from a Wolfenstein Enemy Territory (ET) server based in Melbourne, Australia. Roughly 2.4 million client IP addresses were reduced to a sample set of 330,000 IP addresses, representing clients who played or probed the ET server. We found that 26-28% of client IP addresses could still be pinged directly, we could ping the traceroute-derived last hop router in 62-63% of cases, and in 9-12% of cases we had to use traceroute's own estimate of RTT to the last hop it could find. We also found evidence that traceroute generally reached within one hop of clients who had been logged playing on the server, and within two hops of clients who had simply probed the server.

The obtained RTT and hop-count distributions illustrated the topological and geographical characteristics of clients that played on our Melbourne-based ET server, compared to those who simply probed the server. RTT and hop-count distributions broken down by approximate country of origin also provided an indirect illustration of Australia's challenging topological position for ET game players in the Northern Hemisphere. (Australian players fell between 5 and 15 hops from our server, while international players were well over 15 hops away. Of the clients that played, 60% had an RTT less than 200ms. In comparison, only 10% of people who simply probed our server had an RTT less than 200ms.

Our approach has some distinct limitations. Primarily, one cannot simply assume the Internet's topology is static. RTT and hop-count measurements taken today do not necessarily reflect, in absolute terms, the RTT and hop-count prevailing at the time each client connected to the game server. Peering arrangements may change, and ISPs may move IP address space between their dial-up, cable modem and ADSL access offerings. After-the-fact estimation of RTT and hop counts should be performed as soon as possible after the client IP addresses are collected.

Nevertheless, this paper's technique is a reasonable approach if one primarily wishes to establish a broadly indicative set of RTT and hop-count distributions based solely on client IP addresses found in (game) server logs. It can provide further insights into the RTT tolerance of players by revealing the RTT distributions of clients who probed, but did not play, on a monitored game server.

V.ACKNOWLEDGMENTS

This work was partly supported by the Smart Internet Technology Cooperative Research Centre. http://www.smartinternet.com.au

VI.REFERENCES

[1] G. Armitage, "Sensitivity of Quake3 Players To Network Latency," Poster session, *SIGCOMM Internet Measurement Workshop*, San Francisco, November 2001

[2] T. Henderson, "Latency and user behaviour on a multiplayer game server," *Proceedings of the 3rd International Workshop on Networked Group Communications (NGC)*, London, UK, November 2001

[3] M. Oliveira and T. Henderson, "What online gamers really think of the Internet," *Proceedings of the 2nd Workshop on Network and System Support for Games (NetGames 2003)*, Redwood City, CA, USA, May 2003

[4] T. Henderson, S. Bhati, "Networked games — a QoS sensitive application for QoS-insensitive users?," *ACM SIGCOMM RIPQoS Workshop 2003*, Karlsruhe, Germany, August 2003

[5] T. Beigbeder, R. Coughlan, C. Lusher, J.Plunkett, E. Agu, M. Claypool, "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003," *ACM SIGCOMM 2004 workshop Netgames'04: Network and system support for games*, Portland, USA, August 2004

[6] S. Zander, G. Armitage. "Empirically Measuring the QoS Sensitivity of Interactive Online Game Players". *Australian Telecommunications Networks & Applications Conference (ATNAC)*, Sydney, Australia December 8-10 2004

[7] G.Armitage, M.Claypool, P.Branch. "Networking and Online Games - Undertanding and Engineering Multiplayer Internet Games". John Wiley & Sons, UK, April 2006 (ISBN: 0470018577)

[8] S. Zander, D. Kennedy, G. Armitage. "Dissecting Server-Discovery Traffic Patterns Generated By Multiplayer First Person Shooter Games". *ACM NetGames 2005*, NY, USA, 10-11 October, 2005

[9] C. Jin, H. Wang, and K. G. Shin, "Hop-Count Filtering: An Effective Defense Against Spoofed DoS Traffic," *Proceedings of the 10th ACM International Conference on Computer and Communications Security (CCS)*, pages 30--41, October 2003.

[10] IANA http://www.iana.org/faqs/abuse-faq.htm#SpecialUseAddresses (viewed 30 July 2006)

[11] Maxmind, "GeoLite Country," http://www.maxmind.com/app/geoip_country (viewed 30 July 2006)

[12] Wolfenstein Enemy Territory, http://games.activision.com/games/wolfenstein (viewed 30 July 2006)

[13] K. Auerbach. "Why ICMP Echo (Ping) Is Not Good For Network Measurements". InterWorking Labs, April, 2004 (http://www.iwl.com/Resources/Papers/icmp-echo_print.html, viewed 30 July 2006)

[14] A. Fei, G. Pei, R. Liu, and L. Zhang, "Measurements on delay and hop-count of the Internet," *IEEE GLOBECOM'98 - Internet Mini-Conference*, 1998

[15] GENIUS Project, Centre for Advanced Internet Architectures, http://caia.swin.edu.au/genius (viewed 30 July 2006)

[16] "FreeBSD home page," http://www.freebsd.org (viewed 30 July 2006)