Client RTT and Hop Count Distributions viewed from an Australian ‘Enemy Territory’ Server

Grenville Armitage, Carl Javier, Sebastian Zander
Centre for Advanced Internet Architectures, Technical Report 060223A
Swinburne University of Technology
Melbourne, Australia
{garmitage, cjavier, szander}@swin.edu.au

Abstract–**Network game servers experience traffic caused by actual game players and by remote clients simply probing the game server’s current status. Game clients probe game servers for information such as the current map and number of current players on the server to enable players to find suitable games. The number of clients that probe a given server is orders of magnitudes higher than the number of eventual players. Network level round trip time (delay, or ‘lag’) between a client and server is a very important criterion for players when deciding which server to join. Often the round trip time is roughly proportional to hop count. In this paper we document and investigate the distributions of round trip time and hop count for game clients that only probe and clients that actually play on a public game server. We also examine the geographical distributions of both groups. Our data was gathered from a Wolfenstein Enemy Territory server operating in Melbourne, Australia, in late 2004.**

Keywords-Game Traffic, Round Trip Time, Hop Count

I. Introduction and Previous Work

First Person Shooter (FPS) games are currently a very popular form of multiplayer networked game, typically utilising a client-server model for system communication. Traffic to and from any given FPS game server can be divided into two distinct categories: probe flows and non-probe (game-play) flows. Probe flows occur when a game client requests information about a server whereas game flows are associated with actual game play on the server. 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. In particular, the RTT (colloquially known as ‘lag’) can be a crucial deciding factor in choosing whether or not to play on a particular server [13],[14].

In previous work done by the Centre for Advanced Internet Architectures (CAIA) we discovered 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 [1]. Data was collected from two public Wolfenstein Enemy Territory [2] game servers over 20 weeks in late 2004. One server was located at GrangeNet [3] (Canberra, Australia) and the other was located at CAIA [4] (Melbourne, Australia). The probe and non-probe traffic was analysed for its daily and weekly fluctuations by volume and approximate geographic origin (using GeoIP [10]). 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 CAIA and Grangenet servers. By contrast, game-play accounted for roughly eight thousand flows, 755 million packets and 116 gigabytes of traffic in and out of the CAIA server. (The Grangenet server was less popular and saw far less game-play traffic.)

In this paper we take the analysis further and explore the distribution of client to server RTT and hop count for game clients seen contacting the CAIA server in [1].

Our analysis provides further insight into the geographic and topological distributions of clients who chose to play and those who chose not to play on a 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 test methodology and development limitations. The results are presented in section III. Section IV concludes and outlines future work.

II. Test Methodology

Unfortunately, the data gathered for [1] did not log RTT and hop count information. Consequently our test methodology involved after-the-fact estimation of probable RTTs and hop counts for roughly 2.4 million IP addresses. We needed to work around the fact that client IP addresses logged in late 2004 may now be either completely inactive or no longer associated with an active game client. We chose to make a key assumption - any given IP address is still roughly the same distance away (measured by RTT and hop count) today as it was in late 2004, even if the associated host is no longer the client who played or probed in 2004.

II.1. RTT and Hop Count Estimation

A Python [5] script was developed to cycle through a set of client IP addresses from [1] (those who played and those who simply probed). A mixture of ‛ping′ and ‛traceroute′ estimated the RTT to each destination and measured the hop count (using the TTL of packets coming back from each IP address). Although some concerns have been raised about the suitability of ping for network RTT measurement [6] we decided it was sufficiently accurate for our purposes.

Figure 1 shows the basic probe sequence for a single IP address. 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. Our ping and traceroute probes originated from the same IP subnet that hosted the CAIA server in [1].

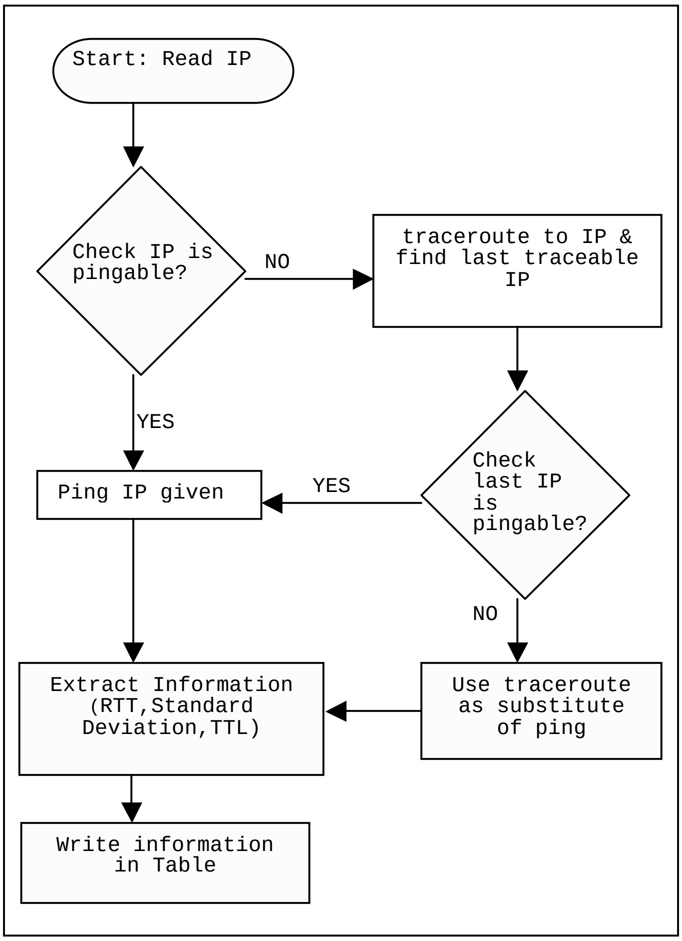
Start: Read IP
Check IP is pingable?
NO
traceroute to IP & find last traceable IP
YES
Check last IP is pingable?
YES
Ping IP given
NO
Extract Information (RTT,Standard Deviation,TTL)
Use traceroute as substitute of ping
Write information in Table

Figure 1: Algorithm for estimating RTT to previously identified IP addresses

To minimise the impact of random network delay fluctuations we sent ten pings (ICMP echo requests) to each destination. The standard deviation of our ten RTT samples indicates whether congestion and route changes may have impacted our RTT estimate during pinging. To avoid inducing congestion or being misinterpreted as a Denial of Service (DoS) attack [7] we sent one packet every two seconds to each specific IP address.

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. We make a simplifying assumption that this 'last hop' is in fact only one hop away from the desired IP destination. (Even though this cannot be generally assumed to be true, our results subsequently suggest the assumption was actually reasonable.)

Hop count is estimated from the TTL field of ICMP messages being returned in response to ping or traceroute. Initial TTL is usually a multiple of 32 [8] and is decremented once at each hop back towards our location. With this knowledge we can use the final TTL to estimate the number of hops traversed by the ICMP response packet.

We explicitly configured traceroute to probe no more than 32 hops away from our location. This substantially reduced the time taken to estimate the last hop of an IP address that could not be pinged directly (since we needed to wait for traceroute to reach its maximum TTL before extracting the identity of the last successfully reported hop).

Along a given route RTT usually increases with increasing hop count. However, different routes may have quite different relationships between RTT and hop count. Physically short hops will contribute far less propagation delay than physically long hops. This becomes particularly evident when we consider a single hop may jump a few metres inside an ISP, or 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 our server.

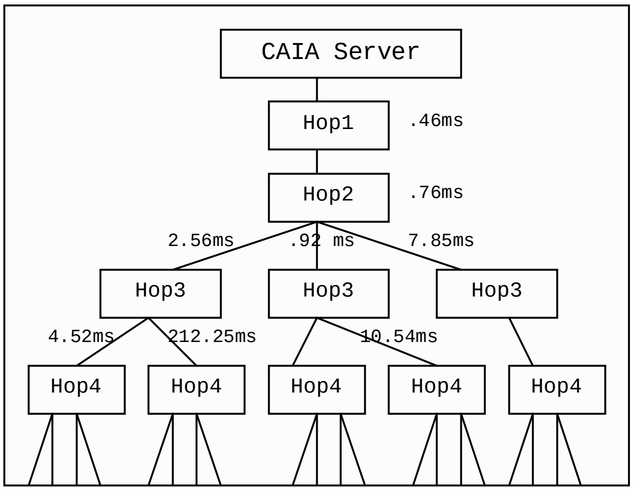

Figure 2: Example of Hops may take by packets as they route to their destination

II.2. Reducing the network scanning period

Pinging every one of 2.4 million IP addresses (from both game flows and probe flows) one by one would take a very long time. We used two techniques to reduce the processing time – selecting subsets of IP addresses to represent larger blocks of IP address space, and pinging multiple IP addresses in parallel.

First we reduced the 2.4 million IP addresses by grouping the IP addresses into unique /24 subnets (covering up to 254 hosts each), and selecting a single IP address to represent all members of the /24. Two assumptions underpin this approach - packets to destinations within a given /24 are likely to take the same path up to the last hop, and the last hop is likely to use the same access technology (and hence have similar latency characteristics) for destinations within the /24. From each /24 subnet a random IP address was chosen to represent that subnet. Table 1 shows the total number of IP addresses compared to the number of IP addresses after this reduction.

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

To actually perform the pinging scans we used an Intel Celeron 2.8Ghz, with 1 GB of RAM and a 60 GB hard drive, running FreeBSD 5.4. On average the ping/traceroute sequence would take 1.45 minutes for a single IP address. We found that 50 scripts could be launched and run in parallel without unduly loading the CPU (the CPU load sat between 3% and 5%, which suggests minimal impact on time-stamping accuracy). Thus our reduced set of IP addresses was further split into 50 non-overlapping subsets, each subset then passed to one of the 50 scripts.

II.3. Running the scripts

Two steps were taken to minimise correlation of ping activity across the concurrent scripts - we launched each of the scripts at random intervals between 0 and 4 seconds, and adjusted the FreeBSD 5.4 kernel's default tick rate from 100Hz to 1000Hz [9]. (With the default 100Hz tick rate we found the pinging activities of many scripts would fall on the same 10ms boundaries, increasing the burstiness of ICMP echo request packet transmission.) Increasing the tick rate also improved the RTT estimation resolution to +/- 1ms.

About 3% of tested IP addresses failed to return proper results the first time each script ran. The script was re-run for such IP addresses to correctly gather the desired information.

To maximise the chances our selected IP addresses would represent an active game client we ran our scripts between Thursdays and Sundays (previously found to be the most popular playing days [1]).

All packet traffic in and out of our machine was logged with tcpdump to enable later calculation of hop counts.

III. Results and Analysis

In this section we will summarise the raw results, discuss the accuracy of our ping estimates, review the use of traceroute to identify the penultimate hop, and describe what our data tells about geographical and topological distribution of both game flow and probe flow clients. Graphs in this section are produced by the "R" statistical analysis program [11].

III.1. Summary of raw results

Our raw results were post-processed to remove anomalous data points before creating the statistics shown in Table 2.

We eliminated data points where the RTT was calculated as being over 1000ms, or the standard deviation for the RTT estimate was over 100ms. Approximately 2.6% of game flow IP addresses and 1.4% probe flow IP addresses were removed for this reason.

Table 2: Game Flow & 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%

A very small number (0.004%) of traceroute-based estimates were eliminated because they returned a private IP address [12] as the last hop.

Finally, where we could not ping an IP address directly we only accepted the traceroute-derived last hop IP address if it was in the same country as the target client IP address (as determined through the GeoIP country database [11]). Only 2% of traceroute-derived data points were removed for failing this test.

III.1. Accuracy of Round Trip Time 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.

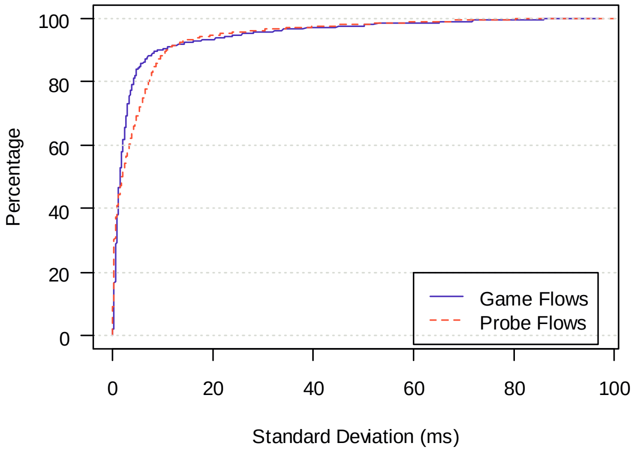

Figure 3: Probe and Game Flow – Standard Deviation of RTT Estimates (CDF)

III.2. Validity of using traceroute to determine 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're 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. The distributions for pingable and non-pingable flows are approximately identical if the non-pingable curve is moved right by one hop (game flows) or two hops (probe flows). 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).

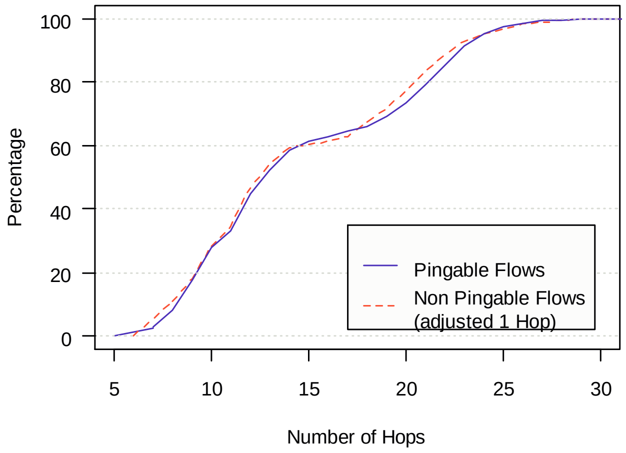

Figure 4: Game Flows – Pingable & Non Pingable Hop Count CDF

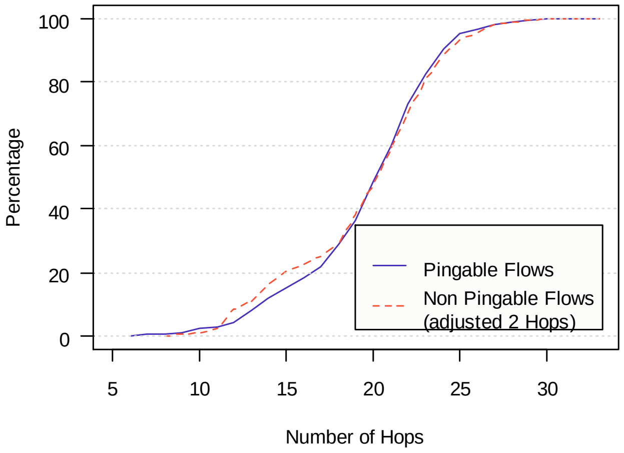

Figure 5: Probe Flows – Pingable & Non Pingable Hop Count CDF

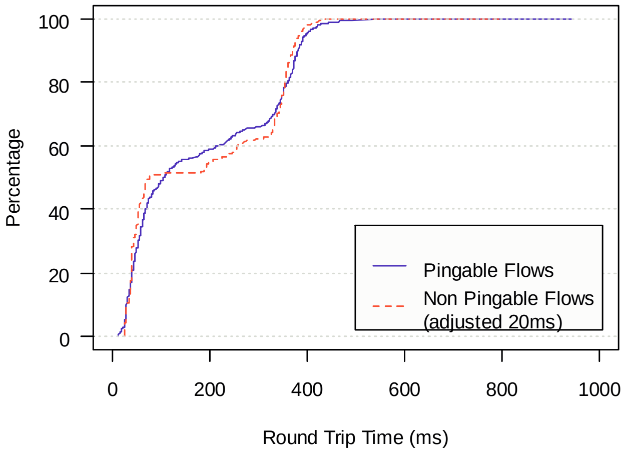

Figure 6: Game Flows – Pingable & Non Pingable Round Trip Time CDF

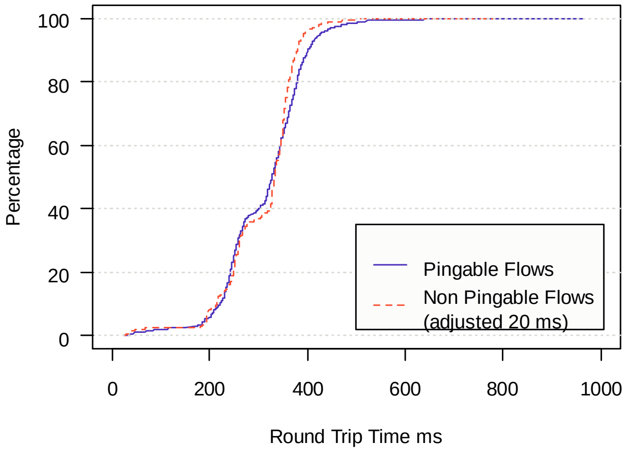

Figure 7: Probe Flows – Pingable & Non Pingable Round Trip Time CDF

(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 and 30ms further away than the ISP router interface we were ultimately able to ping. Thus we appear to be on relatively safe ground in treating the adjusted traceroute-derived data points as equivalent to pingable data points.)

III.3. Geographical Distribution of Game Clients

Using the GeoIP database [10] we identified IP addresses from 54 countries amongst game flows and 138 countries amongst probe flows. However, as shown in Figure 8 a vast majority of game flows were attributable to only a small number of countries. The most prominent is Australia, with 57% of IP addresses. (As noted in [1] our Melbourne-based server′s latency is attractive primarily to Australian players.)

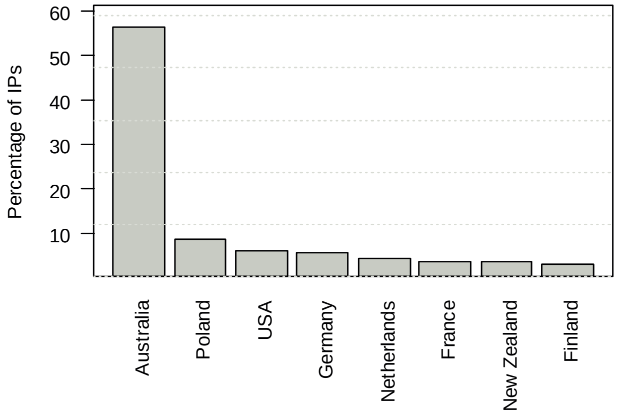

Figure 8: Game Flows – Top 8 Countries

We also see a small but noticeable fraction of game play flows from outside Australia – such as Poland, the USA and Germany. We surmise that these game play flows originated from people interested in our server's particular map sequence or the fact that our server was actively populated at certain times of day (and therefore worth playing on regardless of RTT).

Figure 9 shows that probe flows have quite different demographics (again consistent with [1]). In total European countries contributed to 52% of probe flows, with the USA contributing another 30% of probe flows.

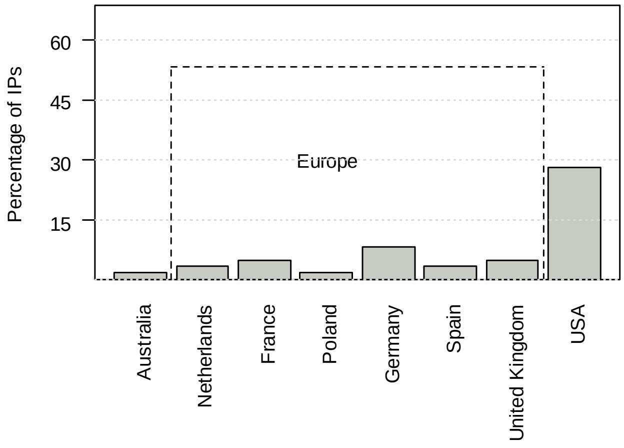

Figure 9: Probe Flows per Country

The topological consequences of being from different countries are revealed in Figure 10 and Figure 11. Figure 10 shows the distribution of hop counts for both game flow and probe flow clients from a number of countries. Australian clients are between 5 and 15 hops from our server while international clients are at least 10 hops away from our server. (As implied by Figure 2, the international clients 10 to 15 hops away are quite likely reached through quite different, physically longer paths compared to the Australian clients who are also between 10 and 15 hops away.)

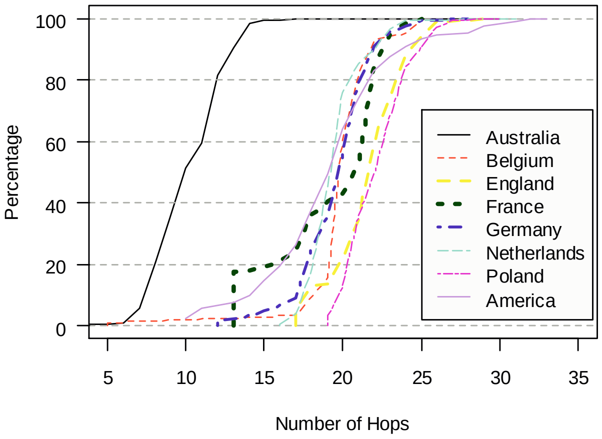

Figure 10: Hop Count CDFs per Country

Figure 11 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 56 ms (with almost all clients being below 100ms) while clients from other countries have RTTs of at least 180-200ms away.

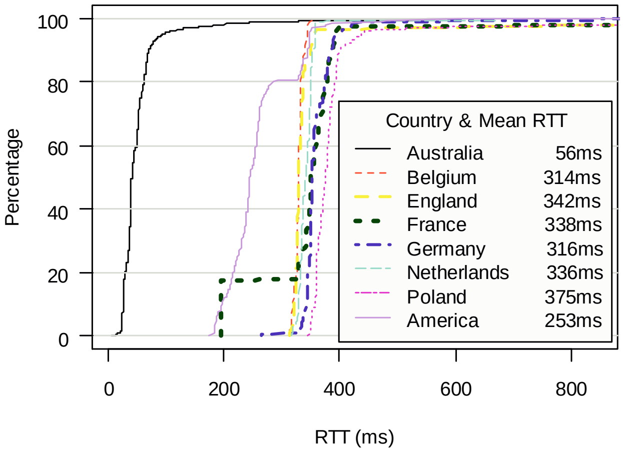

Figure 11: Mean Round Trip Times CDFs per Country

III.4. Round Trip Time & Hop Count Analysis

Comparing the RTT distributions of game and probe flows (as shown in Figure 12) 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 did not play on our server) originate from clients with RTT over

200ms.

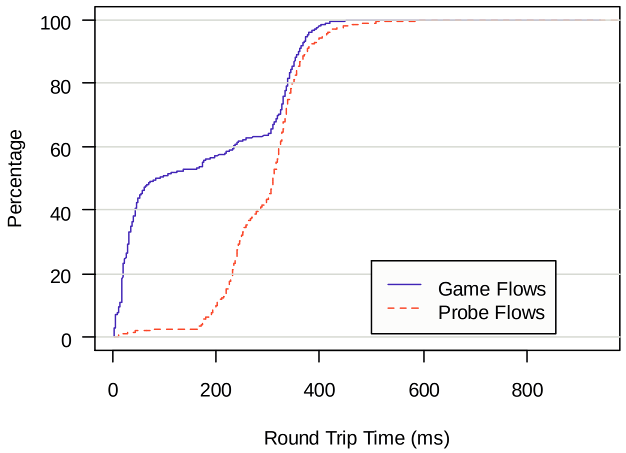

Figure 12: Probe & Game Flows – Round Trip Time CDF

A similar comparison is provided by Figure 13, 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.

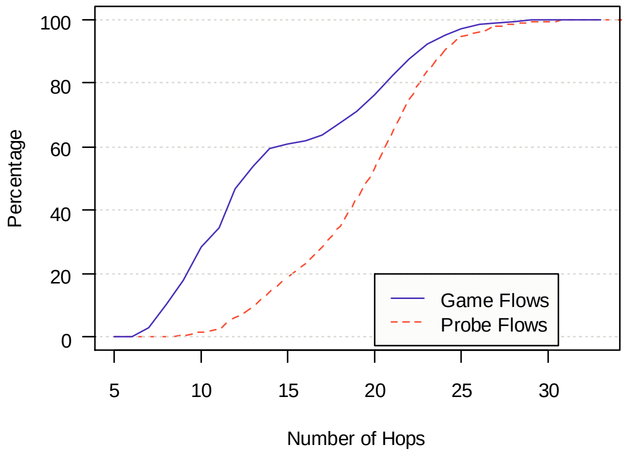

Figure 13: Probe and Game Flows – Hop Count CDF

Figure 14 plots the distribution of hop counts for game flow clients from Figure 13 as a regular histogram. This provides a clear indication that no clients were closer than 5 hops, and that there exists two communities of players – those between 5 and 15 hops away, and those between 17 and 25 hops away.

Figure 15 plots the distribution of hop counts for probe flow clients from Figure 13 as a regular histogram. As we would expect, this distribution is quite different to Figure 14 – the community of probe-only clients is clustered strongly between 10 and 25 hops away from our server. Based on Figure 10 the majority of these probe-only clients (particularly over 15 hops away) reside outside Australia.

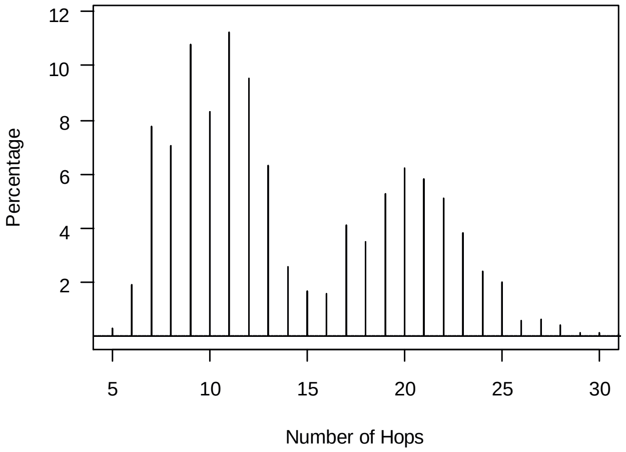

Figure 14: Game Flows – Hop Count Histogram

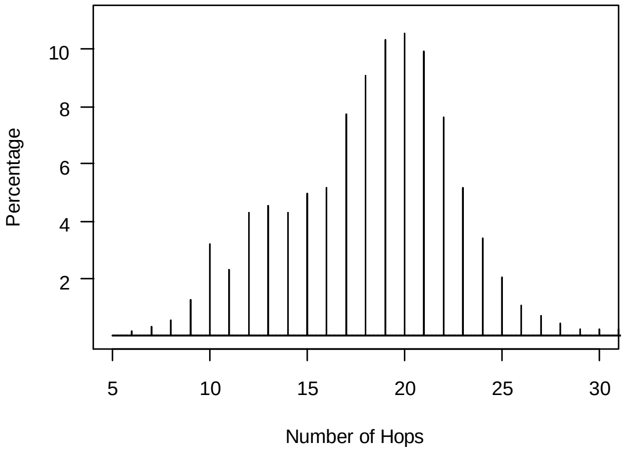

Figure 15: Probe Flows – Hop Count Histogram

Relationships between apparent geographic origin, RTT and hop count are clearly shown in Figure 16 (for game flows) and Figure 17 (for probe flows). Both figures show graphs of average RTT versus hop count for flows originating in five different countries (as determined by GeoIP [10]).

It is clear from both graphs that RTT experienced by players outside Australia are 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's 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.)

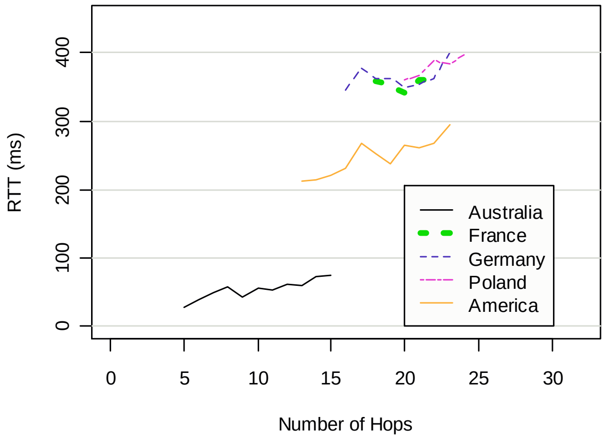

Figure 16: Game Flow – Mean Round Trip Time vs Hops per Country

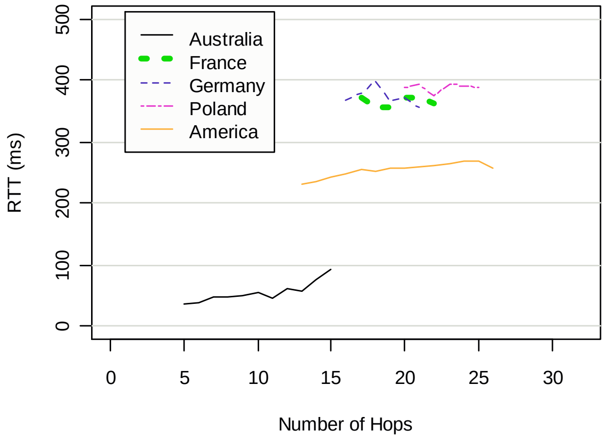

Figure 17: Probe Flow – Mean Round Trip Time vs Hops per Country

IV. Conclusion and Future Work

This paper complements an earlier study [1] on the mix of game play traffic and server probe traffic experienced by a public Enemy Territory game server. We have now, after the fact, measured the round trip times (RTTs) and hop counts to clients of the server analysed in [1]. Our results provide further insights into the RTT and hop count distributions of people who play on a server compared to people who simply probe a server for status information.

In order to obtain RTT and hop count estimates we explicitly pinged a subset of the client IP addresses found in [1]. Many clients no longer existed or had firewalls blocking ping traffic – roughly 72% of game flows and 74% of probe flow could not be pinged directly. In those cases we estimated the RTT and hop count by using traceroute - the last valid IP hop seen by traceroute to a non-pingable IP address was assumed to be one or two hops away from the actual target IP address. This turned out to be a reasonable assumption – validated by comparing the CDF distributions of RTT and hop count for pingable clients and traceroute-derived 'last hops'.

We found that the 57% of clients that played on our server came from Australia and had significantly lower RTT's than other countries. However, we also found a number of clients had played from countries such as Germany and Poland even though they had an average RTT of 300-350ms. The majority of clients that only probed the server were from European countries and the USA.

Australian game play clients fell between 5 and 15 hops from our server, while international players were over 15 hops away. Of the game 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 results also clearly demonstrate the impact of Australia's geographic distance from Europe and America. The first 15 hops towards an international destination show a large RTT jump due to one or more international (long distance) physical layer links. After that, RTT increases with hop count in a more leisurely manner as the path to each client winds its way through local (to the client) ISPs and exchange points.

V. References

[1] 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

[2] "Wolfenstein," http://games.activision.com/games/wolfenstein (viewed 19 December 2005)

[3] GrangeNet, http://www.grangenet.net/advancedcommunications/wp/sim/index.html (viewed on 30th August, 2005)

[4] GENIUS Project, Centre for Advanced Internet Architectures, http://caia.swin.edu.au/genius

[5] Python Programming, www.python.org (viewed on 27th July, 2005)

[6] 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)

[7] Denial Of Service, http://www.cert.org/tech_tips/denial_of_service.html (viewed on 1st August, 2005)

[8] Initial TTL Values, http://members.cox.net/~ndav1/self_published/TTL_values.html (viewed on 5th August , 2005)

[9] FreeBSD home page, www.freebsd.org (viewed on 25th August, 2005)

[10] GeoIP, http://www.maxmind.com/ (viewed 3rd August, 2005)

[11] The R Project for Statistical Computing, http://www.r-project.org (viewed on 12th September, 2005)

[12] IANA http://www.iana.org/faqs/abuse-faq.htm#SpecialUseAddresses (viewed on 30th August, 2005)

[13] S. Zander, G. Armitage "Empirically Measuring the QoS Sensitivity of Interactive Online Game Players", Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia December 8-10 2004

[14] G.J. Armitage, "An Experimental Estimation of Latency Sensitivity In Multiplayer Quake 3", 11th IEEE International Conference on Networks (ICON 2003), Sydney, Australia, September 2003