ABSTRACT
In first person shooter (FPS) games the round trip time (RTT) between a client and server influences player decisions on which server to join. Game servers do not accurately log the RTT of potential clients who only probed the server. We describe a simple, active method of estimating the RTT and hop-count 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. We illustrate using data gathered from a Wolfenstein Enemy Territory server operating in Australia, providing 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.

Categories and Subject Descriptors
C.2.3 [Computer-Communication Networks]: Network Operations - Network monitoring; C.4 [Performance of Systems]: Measurement Techniques

General Terms
Measurement

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

1. INTRODUCTION
First Person Shooter (FPS) games are currently a 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 (or ‘lag’) between a client and server strongly influences enjoyment in such fast-paced interactive games [1][2][3][4][5]. Server operators and Internet service providers (ISPs) can find it useful to characterize the RTT tolerance of clients who frequent their servers. This requires measuring the RTT experienced by clients who probe and join, and clients who probe and never join. FPS servers are usually incapable of logging RTT estimates for clients who probe without joining [6]. An external packet sniffer program such as tcpdump provides only IP addresses.

We describe 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 this technique by estimating the distribution of RTT and hop count for game clients previously seen contacting a Wolfenstein Enemy Territory server based in Melbourne, Australia [7]. The 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. The rest of the paper is organized as follows: section 2 describes the proposed measurement methodology. Section 3 demonstrates the use of this methodology on client IP addresses gathered from a specific game server. Section 4 concludes the paper.

2. RTT AND HOP-COUNT ESTIMATION
2.1 Assumptions
We assume any given IP address is roughly the same distance away today (measured by RTT and hop count) as it was when first logged at the game server. Game clients connect via consumer ISPs whose end-user IP addresses are unlikely to move around topologically. The validity of this assumption degrades over time but should be acceptable over weeks or months.

2.2 Sampling the Client IP Address Set
An active game server can see millions of separate IP addresses over periods of months. We select a subset of logged client IP addresses to represent the characteristics of the path between our

---

1 This is a short version of our original Netgames submission, located at http://caia.swin.edu.au/reports/060801A/CAIA-TR-060801A.pdf
server and all clients. We assume clients under a common CIDR prefix will share much the same path back from the server. (We further assume that IP addresses within a single CIDR prefix are served using a single access technology.) In section 3 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.)

2.3 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.

Our ICMP Echo Requests may 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 such cases we utilize traceroute to probe the path out towards the selected IP address and derive RTT and hop-count estimates. 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.

2.4 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.

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. 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.)

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

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 [8]. 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 [8] 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.

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)

Two sanity checks are applied to the last-hop returned by traceroute. If the reported last-hop is from private address space (e.g. 192.168/16 [9]) or has a different country code than the target client IP address (as reported by a database like GeoLite Country [10]) we exclude this data point from further analysis.

Results from category D are 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.

2.5 Limitations and Considerations
Our RTT measurements are not taken under the same network conditions that existed while each client was accessing the server, and routers do not handle ICMP packets as expeditiously as regular UDP or TCP packets (potentially leading to over-estimation of RTT to the selected client IP addresses [12]).
keep in mind the variable relationship between RTT and hop-count. 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 be thousands of kilometres.

3. ILLUSTRATION USING A GAME SERVER BASED IN AUSTRALIA

3.1 Background

In 2005 we analysed server-probe traffic impacting two ET servers based in Australia [7]. Over a 20-week period each server (in Melbourne and Canberra respectively) saw equal levels of probe traffic (roughly 16 Mflows, 36 Mpackets and 8 Gbytes of data transfer). By contrast, Melbourne’s game-play accounted for roughly 8 Kflows, 755 Mpackets and 116 Gbytes of traffic while the Canberra server saw significantly less game-play traffic. The Melbourne server gave us roughly 2.4 million distinct client IP addresses for which we had no RTT or hop-count information. Neither server had logged its internal RTT estimates for clients who played, nor could they estimate RTTs for clients who simply probed without joining. We failed to keep tcpdump files from which we might extract TTL information to estimate hop-counts.

Table 1: Subnet Reduction of IP Addresses

<table>
<thead>
<tr>
<th></th>
<th>Initial No. Of IP addresses</th>
<th>Reduced No. Of IP addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game Flows</td>
<td>5,469</td>
<td>4,252</td>
</tr>
<tr>
<td>Probe Flows</td>
<td>2,397,879</td>
<td>325,707</td>
</tr>
</tbody>
</table>

We decided to compare the RTT and hop-count distributions of each class of clients. Table 1 shows the result of randomly selecting one IP address from every /24 group to represent the group – from 2.4 million we ended up with roughly 330,000 IP addresses to actively test. Averaged over all IP addresses in category A and category D (section 2.4) the ping/traceroute sequence took 1.45 minutes per address.

3.2 Summary of Raw Results

In Table 2 ‘game flows’ refers to clients who established game-play traffic flows to the server, while ‘probe flows’ refers to clients who established short-lived probe-only flows.

Table 2: Game Flow and Probe Flow Results

<table>
<thead>
<tr>
<th></th>
<th>Game Flows</th>
<th>Probe Flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of IP Addresses</td>
<td>4252</td>
<td>325,707</td>
</tr>
<tr>
<td>Ping directly</td>
<td>28%</td>
<td>26%</td>
</tr>
<tr>
<td>Ping last hop from traceroute</td>
<td>63%</td>
<td>62%</td>
</tr>
<tr>
<td>Used traceroute for RTT computation</td>
<td>9%</td>
<td>12%</td>
</tr>
</tbody>
</table>

More than 90% of the RTT estimates have a standard deviation under 10ms, suggesting the estimation process was fairly consistent over the 10 pings.

3.3 Validity of Using Traceroute to Determine the Last Hop

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’, approximating the desired data point using RTT and hop count to the last hop successfully identified by traceroute. The cumulative distribution functions (CDFs) of measured hop counts for game flow and probe flow IP addresses respectively can be found in [13] (due to space limitations). 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. For RTT estimates the distributions for pingable and non-pingable flows are roughly the same after shifting the non-pingable curve right by 20ms [13]. Consequently, we adjusted all non-pingable data points up by 20ms and one or two hops (for game and probe flows respectively).

3.4 Geographical Distribution of Clients

Using the GeoLite Country database [10] (claimed to be 97% accurate) we identified 54 countries amongst game flows and 138 countries amongst probe flows. 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, European countries contributed to 52% of probe flows, with the USA contributing another 30% of probe flows.

The distribution of hop counts and RTTs for both game flow and probe flow clients from a number of countries can be found in [13] (due to space limitations). Australian clients are 5 to 15 hops away while international clients are at least 10 hops away. 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.

3.5 RTT and Hop Count Analysis

Comparing the RTT distributions of game and probe flows 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. A similar comparison exists for the hop count distributions for game flow and probe flow clients. Less than 10% of probe flows appeared with hop count under 13 (most clustered strongly between 10 and 25 hops away), whereas 60% of game play flows occurred with hop count under 13. CDF plots can be seen in [13]. Average RTT vs hop-count curves broken out by geographic origin (available only in [13] for space reasons) also show that RTT experienced by players outside Australia is dominated by the paths taken just to get to and from Australia itself. Most Australian clients are between 5 and 15 hops away, and less than
monitored game server. RTT distributions of clients who probed, but did not play, a provide insights into the RTT tolerance of players through the solely on client IP addresses found in game server logs, and broadly indicative set of RTT and hop-count distributions based modem and ADSL access offerings. Our technique establishes a and ISPs may move IP address space between their dial-up, cable prevailing at the time each client connected, static, measurements taken today do not necessarily reflect the There are limitations: The Internet's topology is not strictly located ISP equipment racks or Internet exchange points.

4. CONCLUSIONS
It is difficult to log the RTT experienced (or perceived) by clients who simply probe a game server without playing. We describe a technique to establish RTT and hop-count estimates, after the fact, to game clients who may no longer be attached to the Internet. We assume that each client IP address is most likely associated with a consumer Internet connection, and thus assert the RTT and hop-count to every client IP address under a common CIDR prefix will be approximately the same. This step reduces millions of IP addresses to thousands for active probing. As 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. The distance 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. We demonstrate using client IP address data collected from a Wolfenstein Enemy Territory (ET) server based in Melbourne, Australia. Roughly 2.4M client IP addresses were reduced to a sample set of 330K IP addresses, representing clients who played or probed the ET server. We found 26-28% of client IP addresses could 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. We 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 resulting 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. There are limitations: The Internet’s topology is not strictly static, measurements taken today do not necessarily reflect the RTT and hop-count prevailing at the time each client connected, and ISPs may move IP address space between their dial-up, cable modem and ADSL access offerings. Our technique establishes a broadly indicative set of RTT and hop-count distributions based solely on client IP addresses found in game server logs, and provide insights into the RTT tolerance of players through the RTT distributions of clients who probed, but did not play, a monitored game server.

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

6. REFERENCES