Distribution of IP Source Addresses Experienced By Wolfenstein Enemy Territory Game Servers

Grenville Armitage
Centre for Advanced Internet Architectures, Technical Report 070904A
Swinburne University of Technology
Melbourne, Australia
garmitage@swin.edu.au

***Abstract*—Online multiplayer first person shooter (FPS) games typically limit themselves to between 4 and 30+ concurrent players, seemingly limiting the number of source IP addresses seen over time. However, this report demonstrates how common FPS game servers usually 'experience' traffic from hundreds of unique IP source addresses every minute, regardless of an individual game server's popularity or local configuration. The cause is FPS server discovery - a two-step process where clients query a well-known master server for a list of registered game servers, then probe each listed game server in turn. Thousands of clients every day create a continuous 'background noise' of probe traffic toward all registered game servers. Over 13 million probe packets were collected from two 'Wolfenstein Enemy Territory' servers in early 2006. This data was used to characterise the per-minute density of IP source addresses seen by network elements close to game servers. Probes from up to 100 to 550+ unique IP source addresses can be seen every 180 seconds. This report[1] provides some initial insight into the potential memory requirements imposed by probe traffic on network devices that keep per-flow state.**

***Index Terms*—Server discovery, traffic optimisation, network state, measurement**

I. Introduction

Internet-based multiplayer First Person Shooter (FPS) games (such as Quake III Arena [1], Half-Life Counterstrike [2], Wolfenstein Enemy Territory [3] [4], and Half-Life 2 [5]) have become quite common in the past 6+ years. FPS games typically operate in a client-server mode, with game servers being hosted by Internet service providers (ISPs), dedicated game hosting companies and individual enthusiasts. Although individual FPS game servers typically only host from 4 to around 30+ players, there are usually many thousands of individually operated game servers active on the Internet at any given time [6]. This presents a challenge - how do game clients locate up-to-date information about all the servers available at any given time, such that the player can select a suitable server on which to play. Due to the fast-pace and highly interactive nature of FPS games, players seek out game servers that have predictable latency and low packet loss rates. Consequently, a key challenge for those hosting game servers is to understand the impact on their own Internet connection of actually hosting one or more game servers.

Most game traffic research has focused on characterising the network traffic experienced by a game server while people are *actually playing* the game (examples include [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] and [18]). However, FPS game servers also experience a constant 'background noise' of server-discovery traffic from (possibly tens of) thousands of clients around the planet [19].

Server discovery operates similarly for many FPS games (due to the decentralised nature of game server hosting). First, a game client queries a master server unique to the particular game (pre-configured into the game client software). The master server returns a list of hundreds (or thousands) of IP addresses and port numbers representing game servers who have registered themselves as 'active'. The client then steps through this list, probing each listed game server for information (such as the current map type, game type and number of players - typically a brief UDP packet exchange). As a side-effect of this probe the client also estimates the RTT between itself and each game server. All this information is presented to the player (usually as it is gathered), who then selects a game server to join.

Server discovery is usually triggered explicitly by the human player running a particular game client. It may be triggered once or multiple times to refresh the list of available servers presented to the potential player

[1]This technical report is an edited extract from an unsuccessful submission to IMC 2007 on May 10th 2007.

by their client-side server browser. A given client will send out hundreds or thousands of probe packets to find and join only one game server. Consequently, individual game servers end up receiving, and responding to, tens of thousands of probe packets unrelated to the number of people actually playing (or likely to play) at any given time. The background noise due to probe traffic fluctuates over time as game clients around the Internet startup and shutdown.

This paper focuses on the impact of hundreds or thousands of (largely unrelated) game clients independently probing their game's associated game servers, 24 hours a day. In particular, we focus on the spread of IP addresses seen over small periods of time in the vicinity of a game server. Such information may be valuable when sizing memory requirements for network devices that keep per-flow state. (Examples might include NAT or firewall lookup tables in consumer-grade and low-end routers often found deployed 'in front' of FPS game servers. Although UDP is a stateless transport protocol, often such devices will keep some short-term UDP flow state information in order to better support, or track, the end to end applications running on top of UDP - such as games, VoIP, etc.)

Real-world data (over 13 million individual probe packets) was collected from two different 'Wolfenstein Enemy Territory' (ET) game servers in January, April and July 2006. To the author's knowledge, this is the first detailed analysis published of IP source address density over time at multiplayer FPS game servers.

The rest of this report is organised as follows. Section II describes the reason we're interested in IP address density, reviews the specific discovery processes used by ET and summarises the real-world data used in this paper. Section III summarises previous work relating to game server discovery traffic. Section IV discusses the observed probe traffic patterns, and section V reviews the work's implications, limitations and future directions. The report concludes in Section VI.

II. Game Server Discovery

First we introduce the relevance of server discovery traffic to low-end network elements, describe the underlying server discovery mechanisms driving the probe traffic arriving at any given game server, then summarise the real-world probe data used in this paper.

A. The relevance of IP address density

A small percentage of FPS game servers are hosted by commercial companies and ISPs who might be expected to have reasonably powerful routers, firewalls or dedicated NAT boxes in front of their game server machines. However, the business model for online servers in the FPS market has traditionally been to support 'free' self-hosting by enthusiasts. This means a significant percentage of FPS game servers are hosted over modest/low bandwidth links, often sitting behind consumer-grade 'broadband routers'.

Discussion on provisioning of network elements to support online FPS games often revolves around estimating the aggregate capacity (in bits and/or packets per second) required to support a certain number of active players. A game server sitting behind a NAT box or firewall will usually require some form of port-forwarding to be enabled, so that external clients may successfully connect. In principle it should not be necessary to use more than a single UDP port-forwarding rule to handle all client probes coming in from 'the outside'. However, NAT devices are sometimes called upon to exhibit application-level awareness, which leads to the creation and retention of state information relating to every inbound UDP probe. This state information may time out automatically after tens of seconds or a few minutes. Nevertheless it represents a memory requirement that scales in proportion to the number of external FPS clients who come to 'have a look', not just clients who actually join one's server to play.

Consequently, it is a FPS game's distributed server-discovery mechanism that dominates certain types of potential memory consumption in network devices sitting between the FPS game server and the Internet. Whilst game server administrators may limit the number of active players to 4, 8 or so-on, they cannot directly prevent or limit the number of remote clients who may chose to probe their game servers over multi-minute periods of time.

B. Enemy Territory server discovery

ET was released in 2003 as an online-only team-play FPS game, and still has an active online player community. ET is based on the earlier Quake III Arena (Q3A) game engine, and inherits Q3A's underlying server discovery mechanism. Public ET game servers automatically register themselves at etmaster.idsoftware.com, the ET *master server*. This master server becomes a rendezvous point for ET clients around the planet who wish to know what ET game servers are available at any point in time.

Figure 1 illustrates an ET client's server discovery process.

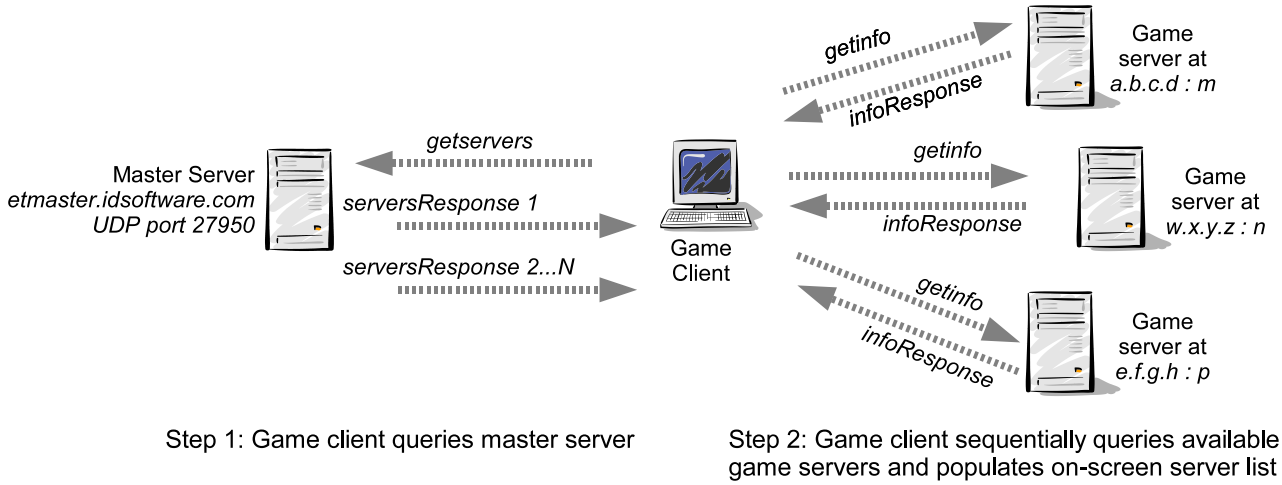

Fig. 1. An ET client's discovery and probing of registered ET game servers

- The client sends a short *getservers* request packet to *etmaster.idsoftware.com* on port 27950, eliciting one or more *getserversResponse* packets (typically well within 2 seconds). These *getserversResponse* packets contain all the currently registered ET game servers. (The ET master server returns multiple *getserversResponse* packets in quick succession when the list of registered game servers is too long to fit in a single UDP payload. A single *getserversResponse* packet can carry IP address:port pairs for up to 112 game servers.)
- After *getserversResponse* packets are received the game client begins issuing *getinfo* probes to each listed game server. Game servers are probed in the order in which they were listed in the master server's *getserversResponse* packet(s).
- Each game server's reply comes back in an *infoResponse* packet. The game client populates its on-screen 'server browser' using information contained in each *infoResponse* packet and the game server's round trip time (RTT, estimated from the time between sending a *getinfo* and receiving a matching *infoResponse*).
- At any time during (or after) the *getinfo* / *infoResponse* process the player may chose a specific game server to play on from the information presented in the onscreen server browser.

As a small optimisation, an ET client partially overlaps the reception of *getserversResponse* packets and the emission of *getinfo* probes. The first 16 game servers are probed in sequence as soon as the first *getserversResponse* packet arrives from the master server, with additional *getinfo* probes sent as previous probes are answered. No more than 16 probes remain outstanding (unanswered) at any one time.

From an ET game server's perspective, within minutes of registering with the master server it will begin seeing an influx of *getinfo* probe packets. These probes come from active ET clients around the Internet (Figure 2) and automated game server monitoring systems (such as ServerSpy [20]). The flow of player-triggered probe traffic will fluctuate with a 24-hour period [19], but rarely ever stop while the ET game server remains registered.

After establishing basic information about all registered game servers, the player may discover additional information about a particular game server by pressing the "Server information" button in the client's server selection browser. This triggers a *getstatus* request to the selected game server, eliciting additional information about the selected game server in a *statusResponse* reply. (For this report we ignore *getstatus* messages as they do not materially increase the number of unique IP addresses seen by the game server over multi-minute time intervals.)

C. Collecting real-world probe traffic

During late 2005 and 2006 two identically-configured ET game servers were monitored 24 hours a day, 7 days a week in Australia. One physical server was located in Melbourne, Australia (gs.caia.swin.edu.au at the Centre for Advanced Internet Architecture, CAIA) and the other in Canberra, Australia (gs.act.grangenet.net, hosted by Grangenet [21]). Each physical server hosted one ET server on UDP port 27961. Both game servers were registered with the ET master server and open for public

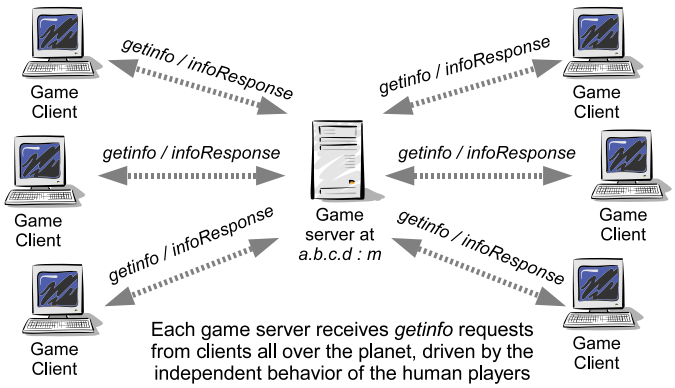

Fig. 2. *getinfo* probes come from clients all around the world

play. Every packet entering and leaving both servers was captured using tcpdump [22] for subsequent analysis.

Both the CAIA and Grangenet servers used 100Mbit/sec LAN interfaces and were connected to the Internet over relatively uncongested academic research links. (Unfortunately, the servers were not always stable, so certain months had incomplete collections of data.) Table I shows the total number of *getinfo* packets received by each ET server in January, April and July 2006. ("(-)" indicates an incomplete collection of probe traffic that month.)

	Jan 2006	Apr 2006	Jul 2006
CAIA	3219842	2740068	2399319
Grangenet	(-)	2835542	2492274

TABLE I
NUMBER OF ET *getinfo* PROBES SEEN EACH MONTH

MaxMind's free GeoLite Country database [23] was used to differentiate probe traffic by source country. MaxMind claims this database correctly maps 97% of all IP address allocations to country codes. They also have significant coverage of the active IP address space (for example, of the 8.26M probes seen across all three months at the CAIA ET server, only 2353 could not be resolved to a country). Table II ranks the top five countries probing the CAIA ET server (and the number of probes sent) in January, April and July 2006. A similar distribution was seen on the Grangenet server. (ET is set in World War II, where players choose to play either as Allied or Axis soldiers. This may underly the apparent interest in this game by European players.)

III. RELATED WORK

It does not appear that any prior work has explored the dynamic variations in IP address density experienced by game servers due to server discovery probe traffic.

	Jan 2006	Apr 2006	Jul 2006
Number of countries	127	125	125
1st	PL 502501	PL 447632	PL 379635
2nd	US 446599	US 365418	US 363581
3rd	DE 322553	DE 307344	DE 264453
4th	FR 235594	FR 223691	FR 183806
5th	NL 217775	NL 190035	NL 144541

TABLE II
TOP 5 COUNTRIES PROBING CAIA'S ET SERVER EACH MONTH

In 2003 Chambers et al. [24] proposed dynamic server re-discovery, redirecting players from one game server to a closer game server based on inferring geographic locality from client IP addresses. The characteristics of server discovery traffic itself were not addressed.

The ET servers used in this paper were the basis for an earlier study of the relationship beteween game-play (clients connected and playing) and probe traffic impacting on public game servers. In 2005 Zander et al. observed that aggregate game-play traffic and server discovery probe traffic both exhibited similar, yet out-of-phase, 24-hour cycles [19]. Probe traffic would rise and fall as the number of *potential* players changed over time, whereas game-play traffic depended only on the number of *actual* people who chose to play on our servers. Typically this latter group (actual players) was made up of people geographically close to our servers, whereas the former group (potential players) was dominated by the 24-hour cycle of players from Europe and the United States of America. (Few people played on the Grangenet server, yet it saw very similar levels of probe traffic to that seen by the CAIA server). A break-down of traffic by geographical region confirmed each region had its own 24-hour cycles. Probe and game-play activity peaked during the evenings of each region's specific timezone. However, [19] did not provide any detailed information on source IP address density variations over sub-hour intervals.

In 2006 Armitage et al. [25] observed that every 36 minutes the ET master server would rotate the rank of every game server within the list returned in step 2 of Figure 1. Over 36 minutes a given game server would find itself 1st, 2nd.... Nth in the list returned by the master server. The actual ordering of game servers from the ET master server appeared entirely unrelated to the client's topological relationship to either the master server or the active game servers. Client-side optimisations to Figure 1's server probe algorithm were

proposed to reduce the number of probes sent by a given client. However, the impact on the range of IP source addresses seen by each active game server over time was not addressed.

IV. Observed probe traffic

A. The human origins of most probe traffic

Before considering the distribution of IP source addresses seen over time it is worth reviewing the evidence that most probe traffic is triggered by human players. Ultimately this pattern influences the peak and daily variation in unique IP addresses seen by an individual game server over time.

Figure 3 shows the average number of ET probes seen per hour over a typical 24-hour period from all source addresses. CAIA's ET server statistics are shown for the months of January, April and July 2006. The Grangenet ET server's statistics are shown for April 2006. The x-axis is in 'hours relative to GMT+10:00', where 0 is midnight and 23 is 11pm in the GMT+10:00 timezone (Eastern Standard Time for both ET servers).

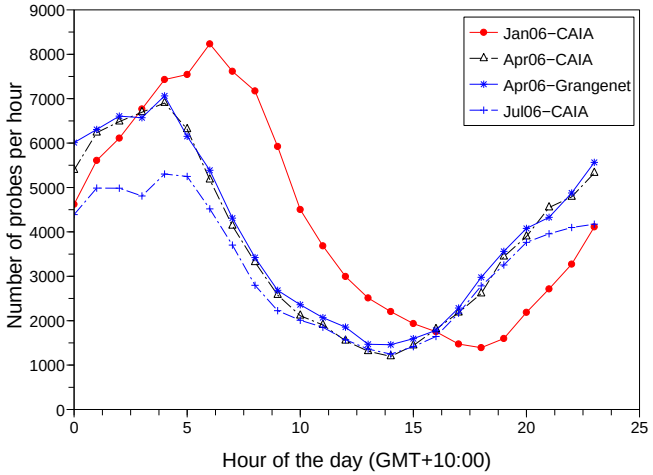

Fig. 3. Average number of ET *getinfo* packets per hour of a day in January, April and July 2006

Focusing on the CAIA ET server, Figure 4 shows the average number of probes per hour of the calendar week during April 2006 from the top two sources of probes that month - the United States (US) and Poland (PL) - and from Australia (AU) for comparison with clients closer to the GMT+10:00 timezone. The 24-hour cycle is quite evident, as is a distinct phase difference between PL, US and AU distributions. The x-axis is in 'hours relative to GMT+10:00', where 0 is midnight on Sunday and 167 is 11pm the following Saturday in the GMT+10:00 timezone.

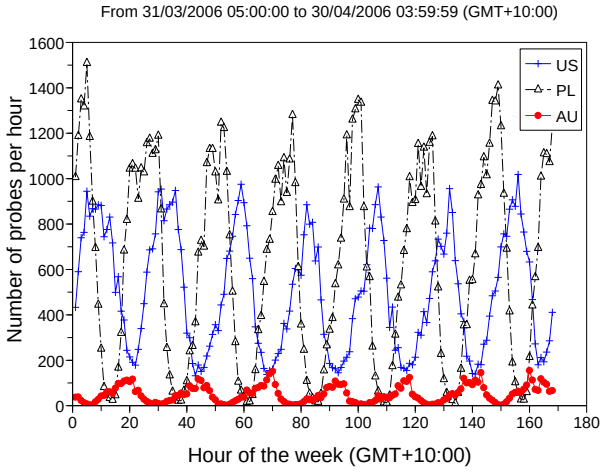

Fig. 4. Average number of ET *getinfo* packets per hour of a week in April 2006 from Poland, the USA and Australia

Figures 3 and 4 seem to confirm the belief that ET server-discovery probe traffic is largely a human-initiated process. The aggregate traffic seen in Figure 3 is made up of regionally-specific probe traffic, which Figure 4 shows is strongly influenced by the times of day that potential players are available from each geographic region. The offset of Figure 3's January 2006 curve may also be attributed to human factors behind the probe traffic. Most of April 2006 and all of July 2006 were 'summer time' in the northern hemisphere (from where the majority of probes originate), with increasingly longer days and shorter nights. January 2006 was 'standard time' (and the middle of winter). Not surprisingly the peak probing hour (relative to GMT+10:00) has shifted roughly 1 to 2 hours between wintertime and summertime because of 'daylight saving time' and players simply modifying their daily patterns of life. Finally, Figure 3 shows that both the CAIA and Grangenet ET servers saw almost identical levels of probe traffic (despite the CAIA server being far more popular with actual players [19]).

B. IP source addresses

Figure 5 is a variant on Figure 4, this time with the average number of *unique* IP source addresses seen per hour of the week. (Multiple packets from the same source within the same hour count only once in Figure 5.) Based on the 24-hour cycles in Figure 5 it seems reasonable to conclude that the number of probes per hour seen in Figure 4 owes its fluctuations to increasing and decreasing diversity in the set of clients (IP endpoints) actually sending probes. Broadly speaking, we see between 1.5 and 2 probes per unique IP address per

hour.

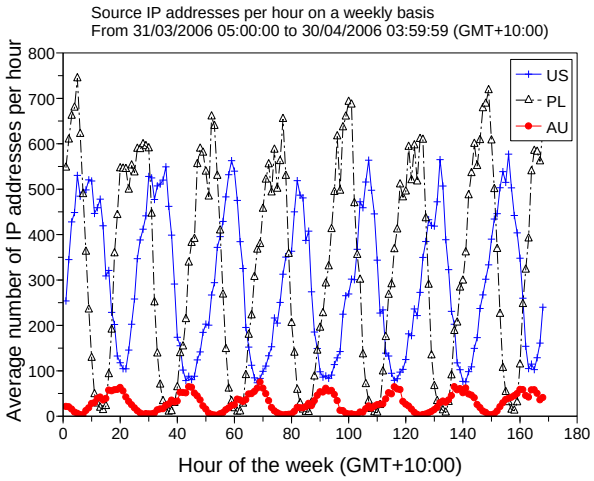

Fig. 5. Number of unique IP addresses seen per hour of a week in April 2006 from the USA, Poland and Australia (CAIA ET server)

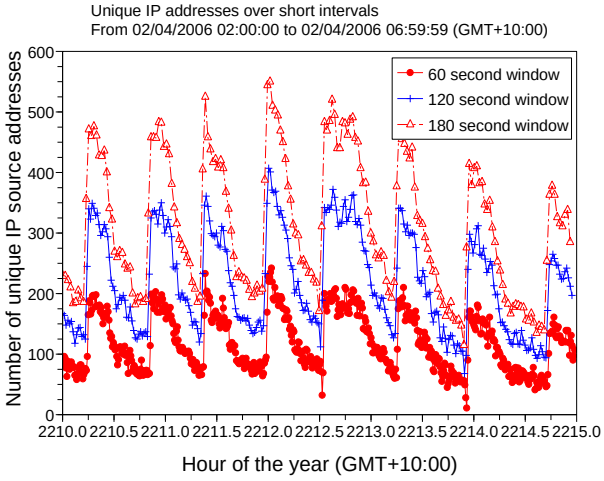

Fig. 6. Number of unique IP addresses seen by the CAIA ET server across 60, 120 and 180 second intervals around the peak hour in April 2006

While Figure 5 is of interest if your network elements (NAT lookup tables, etc) have idle timeouts in the order of an hour or more, many systems are likely to consider flushing unused look-up table entries over periods of a few minutes. Figure 6 provides a close-up view of the number of unique IP addresses (from all regions) seen by the CAIA ET server over the busiest 5-hour period in April 2006. Each data point is calculated by summing the number of unique IP addresses seen over the next N seconds (for N = 60, 120 and 180). We then moved forward N/2 seconds in time, and recalculate. In other words, if your network element implemented a 60-second idle timeout, you would see up to 200 unique IP addresses in any given 60-second period. If you implemented a 180-second idle timeout, you would need to allow for up to 550 unique IP addresses in any given 180-second period. Figure 6 also suggests that probing over short intervals (less than 180 seconds) is largely performed by independent clients (as the number of unique IP addresses in the 180 second window is almost 3 times that in the 60 second window).

(Note that despite the similar range of y-axis values, Figure 6 evaluates the aggregate traffic from all around the planet during the busiest 5-hour period of April 2006, whereas Figure 5 considers region-specific averages over multiple weeks in the month of April 2006.)

Figure 6 also appears to support a hypothesis (expressed in [25]) that human players might often terminate their client's server discovery process before probing all the game servers returned by the master server. (For example, the player has found a suitable game server, and decided to play now rather than wait for Figure 1's client-side server discovery process to complete.) Figure 7 is reproduced from [25] and shows the ET master server cycling three registered game servers from top to bottom of its *serversResponse* list every 36 minutes. The visual similarities between Figure 6 and Figure 7 suggest that the number of unique IP addresses seen by the game server is influenced by the game server's rank within the ET master server's *serversResponse* list.

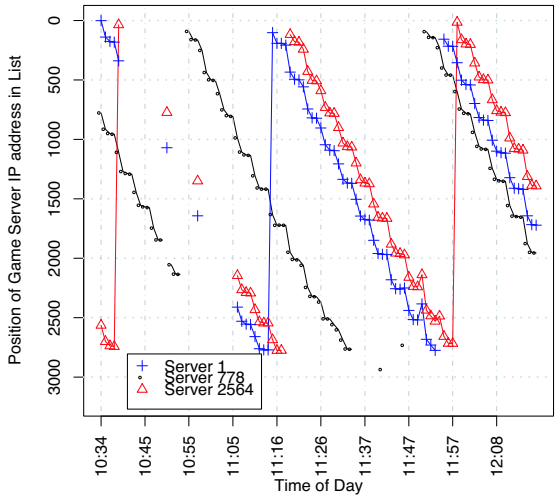

Fig. 7. ET master server rankings for 3 specific game servers fluctuates every 36 minutes

V. Discussion and Future work

A. An undercurrent of automated probing

Although server-discovery probe traffic is primarily driven by human activity, Figure 4 reveals that not every country 'goes to sleep'. For example, the probe traffic from Poland drops almost to zero once every 24 hours, whilst the United States continues to generate some base level of probe traffic at all hours of the day.

Manual inspection of the raw traffic reveals that a very small number of probe sources are automated systems, regularly and continuously polling game servers all around the world. These systems (such as ServerSpy [20]) are designed to create rankings of players across different genres of FPS games, and across all public servers for particular FPS genres. It is likely that such services contribute to the non-zero base level of US traffic in Figure 4. (The US west coast and east coast both share a number of hours in the early morning where people would normally be asleep, and it seems unlikely that Polish players are less dedicated than US players.)

Nevertheless, the number sources generating automated probes is sufficiently low that the short-term density of unique IP addresses over time is dominated by the geographical diversity of human players.

B. Trends over multiple months

It was noted in [19] that the peak probe activity is likely proportional to the density of potential players around the planet. Figure 3 suggests the total number of ET clients being turned on and off each day was slowly declining during the first half of 2006. Figure 8 makes this trend much clearer, plotting the actual number of probes per 'hour of the year' at the CAIA server over three separate months. The peak probes-per-hour on any given day is clearly declining as the year progresses.

It is important to note that Figure 8's decline in probe levels is not a function of the CAIA game server's own popularity. There are simply less hosts on average sending out server-discovery probe packets during July 2006 than January 2006. (If anything, this should be interpreted as a drop-off in ET's overall popularity around the world.) Conversely, if the number of ET players had been increasing, we would see an increase in probe traffic over time. The sources of long-term fluctuations in peak probe traffic are beyond the scope of this paper. Nevertheless, any long-term prediction of the density of unique IP addresses due to ET players would do well to consider such trends.

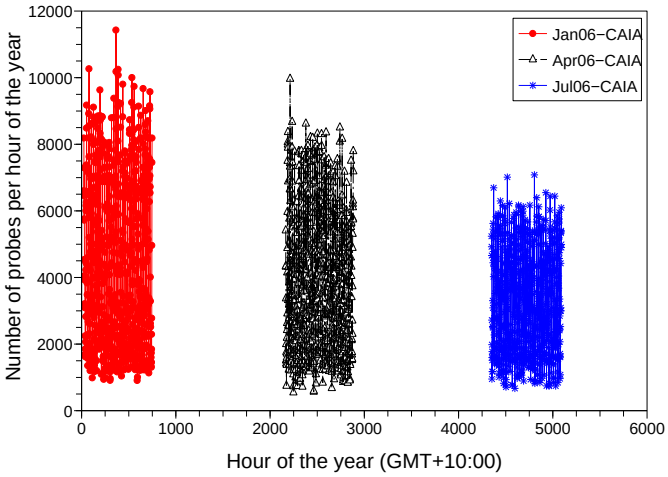

Fig. 8. Number of probes per hour of the year at the CAIA ET server

C. Master server rankings and client-side optimisations

While measuring the probe traffic received by two sites in Australia we make an assumption that ET and Steam server-discovery does not bias toward or against game servers who are closer to, or further from, the querying game client. In the case of ET, this assumption seems reasonably warranted - it is a simple discovery protocol, derived from Quake III Arena (and apparently inherited by Quake 4's online multiplayer system). However, proposals have been made for optimisations to the ET server discovery process (e.g. [25]) that would reduce the number of probes sent (and hence the network bandwidth consumed by) clients who are simply trying to find a 'good' server on which to play. At an abstract level, such optimisations boil down to guiding the client's probing toward game servers more likely to satisfy the potential player's interests (based on some locally defined metric, e.g. minimising the likely RTT to the target game server).

Deployment of such optimisations would, naturally, skew the probe traffic seen by any one game server. Figures 6 and 7 show that for ET simply dropping a game server's rank in the master server's list can cause a drop in probe traffic received by that individual game server. A client-side algorithm that probed game servers in order of likely RTT would create a similar drop in the probe traffic between clients and game servers 'too far away' from each other for decent game play. Game servers far from the center of mass of potential players (such as ET servers located in Australia) would see disproportionately less probe traffic than servers closer to the majority of potential players.

VI. Conclusion

Running an online FPS game server will immediately cause a steady stream of inbound probe packets to arrive day and night. This probe traffic occurs regardless of your game server's popularity or any internally configured limits on the number of players allowed to actually *play* at any given point in time.

The consequences are of interest when scaling the memory requirements of network elements that are often placed 'in front' of game servers. A small percentage of FPS game servers are hosted by commercial companies and ISPs who might be expected to have reasonably designed routers, firewalls or dedicated NAT boxes in front of their game server hosts. However, the vast majority of FPS game servers are hosted by enthusiasts, often over low bandwidth links and sitting behind consumer-grade 'broadband routers', using NAT with port-forwarding enabled so that their public game server may be probed. NAT devices that attempt some form of application-awareness will tend to keep per-flow state relating to UDP traffic. Consequently, FPS server-discovery can have a notable impact on state information tracked by low-end NAT devices, beyond the control of any individual game server administrator.

This paper has looked at one modestly popular online multiplayer FPS game - Wolfenstein Enemy Territory. It seems clear that flow-aware network devices situated 'in front' of individual ET game servers would expect to see from 100 to 550+ unique IP source addresses over multi-minute intervals.

It is relatively easy to replicate this research. Setup one's own public game server, and capture all the probe traffic that begins to arrive over time. However, future optimisations in the client-side server discovery mechanisms may skew the probe traffic received by any particular game server. It would be advantageous to replicate this research using game servers located in a more diverse set of locations around the planet.

References

[1] id Software, *Quake III Arena*, http://www.idsoftware.com/, as of April 29th 2007.

[2] Valve Corporation, *CounterStrike: Source*, http://counter-strike.net/, as of April 29th 2007.

[3] id Software, *Wolfenstein Enemy Territory*, under "Downloads" at http://www.enemyterritory.com/main.html, as of April 29th 2007.

[4] (fan site), *Wolfenstein Enemy Territory*, http://www.enemy-territory.com/, as of April 29th 2007.

[5] Valve Corporation, *Half-Life 2*, http://half-life2.com/, as of April 29th 2007.

[6] G. Armitage, M. Claypool, and P. Branch, *Networking and Online Games - Understanding and Engineering Multiplayer Internet Games*. United Kingdom: John Wiley & Sons, April 2006.

[7] M. Borella, "Source models of network game traffic," *Computer Communications*, vol. 23, no. 3, pp. 403–410, February 2000.

[8] T. Henderson and S. Bhatti, "Modelling user behaviour in networked games," in *9th ACM International Conference on Multimedia (ACM Multimedia)*, 2001.

[9] G. Armitage, "An experimental estimation of latency sensitivity in multiplayer Quake3," in *Proceedings of 11th IEEE International Conference on Networks (ICON)*, September 2003.

[10] W.-C. Feng, F. Chang, W.-C. Feng, and J. Walpole, "Provisioning on-line games: A traffic analysis of a busy Counter-Strike server," in *SIGCOMM Internet Measurement Workshop*, 2002.

[11] T. Lang, G. Armitage, P. Branch, and H.-Y. Choo, "A synthetic traffic model for Half-Life," in *Australian Telecommunications, Networks and Applications Conference (ATNAC)*, December 2003.

[12] J. Farber, "Traffic modelling for fast action network games," *Multimedia Tools and Applications*, vol. 23, pp. 31–46, December 2004.

[13] T. Lang, P. Branch, and G. Armitage, "A synthetic model for Quake III traffic," in *Advances in Computer Entertainment (ACE2004)*, June 2004.

[14] W.-C. Feng, F. Chang, W.-C. Feng, and J. Walpole, "A traffic charaterization of popular on-line games," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, pp. 488–500, June 2005.

[15] S. Zander and G. Armitage, "A traffic model for the XBOX game Halo 2," in *15th ACM International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV2005)*, June 2005.

[16] C. Chambers, W.-C. Feng, S. Sahu, and D. Saha, "Measurement based characterization of a collection of on-line games," in *Internet Measurement Conference 2005 (IMC2005)*, October 2005.

[17] P. Branch and G. Armitage, "Extrapolating server to client ip traffic from empirical measurements of first person shooter games," in *5th Workshop on Network System Support for Games 2006 (Netgames2006)*, October 2006.

[18] ——, "Measuring the auto-correlation of server to client traffic in first person shooter games," in *Australian Telecommunications, Network and Applications Conference (ATNAC)*, December 2006.

[19] S. Zander, D. Kennedy, and G. Armitage, "Dissecting server-discovery traffic patterns generated by multiplayer first person shooter games," in *Proceedings of ACM Networks and System Support for Games (NetGames) Workshop*, October 2005.

[20] *ServerSpy - Online PC Gaming Statistics*, http://www.serverspy.net/, as of April 27th 2007.

[21] *GrangeNet*, http://www.grangenet.net/, as of April 27th 2007.

[22] *TCPDUMP public repository*, http://www.tcpdump.org/, as of April 27th 2007.

[23] MaxMind, *GeoLite Country*, http://www.maxmind.com/app/geoip_country, as of April 29th 2006.

[24] C. Chambers, W.-C. Feng, F. W.-C., and D. Saha, "A geographic, redirection service for on-line games," in *ACM Multimedia 2003 (short paper)*, November 2003.

[25] G. Armitage, C. Javier, and S. Zander, "Topological optimisation for online first person shooter game server discovery," in *Proceedings of Australian Telecommunications and Network Application Conference (ATNAC)*, December 2006.