Measurement and Estimation of Network QoS Among Peer Xbox 360 Game Players

Youngki Lee[1], Sharad Agarwal[2], Chris Butcher[3], and Jitu Padhye[2]

[1] KAIST
[2] Microsoft Research
[3] Bungie Studios

1 Introduction

The research community has proposed several techniques for estimating the quality of network paths in terms of delay and capacity. However, few techniques have been studied in the context of large deployed applications. Network gaming is an application that is extremely sensitive to network path quality [1,2,3]. Yet, the quality of network paths among players of large, wide-area games and techniques for estimating it have not received much attention from the research community.

Network games broadly fall into two categories. In some games (e.g. MMORPGs, web-based casual games, Quake) with a client-server architecture, players communicate with a large, *well-provisioned*, and *dedicated* game server [4,5]. In some games with a peer-to-peer (P2P) architecture, players communicate with each other directly or via a dynamically chosen peer at some player's house. In Ghost Recon, Halo series, and others for the Xbox and Xbox 360 consoles, a server assists players in discovering other peers to host and play with.

Accurate and scalable estimation of the network path quality (NPQ) between peer game players is especially critical for games with a P2P architecture. These players need to have good network connectivity to each other, so accurate NPQ data is essential for "matchmaking" - i.e. to determine which players should play with each other. Furthermore, NPQ estimation needs to be done in a scalable manner. If the number of peers is large, it may not be not feasible to probe all of them.

Prior research on P2P games has used data from only a small number of players [6]. We study a much larger data set, from Halo 3 : a popular Xbox 360 game. We cover 5.6 million unique IP addresses that played 39.8 million game sessions in 50 days. Peers in each game session gather NPQ data and report it to the central Xbox server for matchmaking purposes.

This paper makes the following contributions:

- We present data from a large P2P gaming application. The population is several orders of magnitude larger, and far more geographically diverse than any previously reported study. Given the number and geographical diversity of players, we consider this to also be a large-scale study of path quality over the wide-area Internet.
- We study temporal and geographical correlations in the NPQ data, and propose three different predictors that can provide a rough estimate of NPQ between a pair of players, without requiring any probing. There can be millions of game players

M. Claypool and S. Uhlig (Eds.): PAM 2008, LNCS 4979, pp. 41–50, 2008.

on-line at any time, and any techniques that can avoid having to perform network probes between all of them can not only reduce network overhead but also reduce the amount of time players have to wait before starting a game over the Internet.

2 Background

The Microsoft Xbox 360 game console supports on-line game play with the Xbox Live subscription service. The Halo series of First Person Shooter (FPS) games has sold over 15 million copies worldwide. We focus on the latest edition, Halo 3. Each Halo 3 Internet game session can include up to 16 players. One console in each game session is selected to be the game host or server. All game communication between other players is relayed through this host console. The Xbox Live server provides accounting and matchmaking services. Therefore, the NPQ between consoles and the Xbox Live server is less important to the overall gaming experience than the NPQ between the consoles themselves. An "excellent" Halo 3 experience has under 50ms of latency and 50Kbps to 70Kbps of bandwidth between each client console and the host console. Note that the host console may consume up to 1Mbps ((16-1)*70Kbps) of bandwidth. A "good" experience can be achieved with 150ms latency and 30Kbps of bandwidth. Hence, it is important to group consoles so that they each have good NPQ to the host console. This is critical in this architecture because the host is a fellow player's console, typically on a consumer broadband connection, and not a well provisioned, dedicated server.

The Xbox Live server helps with "matchmaking" - setting up such groups, of up to 16 players, from among the hundreds of thousands on-line at any time. A player who starts an Internet game session will sign on to the Xbox Live service and run Halo 3 on her console. She will select Internet game play and can specify several criteria for the session, such as the type of game (e.g. free for all or team objective). With some probability, this console will become a peer game host, instead of a game client. This probability depends on the chosen type of game. If the console is a game host, it will wait for other consoles to discover it, probe their NPQ to it, and join the game.

If it is a game client, Xbox Live will send it IP addresses for the other consoles on the Internet that are hosting games of the specified type. This console will send and receive several packet pairs with each IP address. The Xbox 360 networking stack implements the standard packet pair estimation technique [7]. Packet pairs are performed serially and do not overlap with each other. The console will then have an estimate of the round-trip latency (RTT), and the upstream and downstream network capacity with each candidate game host. While being very lightweight, packet pair measures bottleneck link capacity but not available bandwidth. These values are logged by the Xbox Live service. The user is shown those host consoles that it has the best NPQ to. For conciseness, we leave out several details such as NAT traversal.

Little is known about the population of on-line P2P game players. Their geographic diversity, diurnal behavior, typical network delay and capacity are useful parameters to network models of game systems for future research. This information can help build estimators of NPQ between any two game consoles on the Internet. Even merely identifying the pairs of consoles with extremely poor NPQ can significantly reduce the total number of probes, thereby reducing network overhead and user wait time.

3 Data and Methodology

Xbox Live stores information about every Internet game session for Halo 3. In a typical week ending on 29 January 2008, we find that 72.5% of Internet game sessions required matchmaking; when weighted to account for players per game, it is 83.5%. By a "session", we mean an attempt to search for an Internet game - the user may have eventually joined a game or decided not to. The log has the UTC time and the public IP address of the console searching for a game. This console may have probed several other consoles that were hosting games of the requested type - for each probe to a candidate host console, we have the host IP address, median round trip time (RTT), and average capacities upstream to host and downstream from host. We use the term "probe" to mean 4 packet pair tests from the client console to a host console and 4 in the reverse direction. We use "player", "console" and "IP address" interchangeably.

Table 1. Data sets

Halo 3 Phase	Start	End	Distinct IPs	Matchmaking games	Hosts probed
Internal alpha	11/30/2006	01/23/2007	4,025	314,606	207,595
Internal beta	05/08/2007	05/21/2007	732,487	20,747,695	33,338,060
Public beta	05/22/2007	06/11/2007	903,782	23,182,323	38,453,035
Release	11/14/2007	01/03/2008	5,658,951	39,803,350	126,085,887

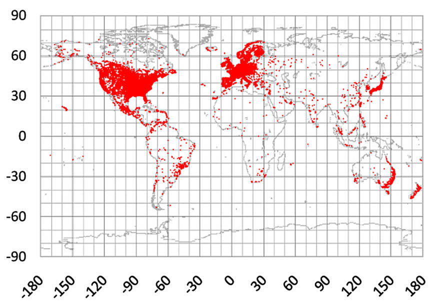

Fig. 1. Geographic distribution of players

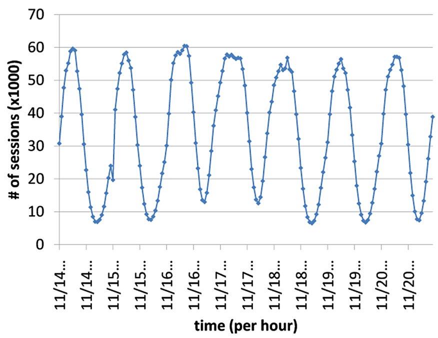

Fig. 2. Game sessions per hour

Table 1 lists our data sets. For conciseness, we focus on the "Release" data set for Halo 3. Due to the extremely large number of game plays, we limit the data set in two ways - we consider a 50 day period and we only consider a randomly selected 20% of the matchmaking game sessions. The resulting data set covers over 126 million probes between over 5.6 million IP addresses. For geographic analysis, we use the commercial MaxMind GeoIP City Database from June 2007. It was able to provide the latitude and longitude for over 98% of these IP addresses.

4 Player Population Characterization

In this section, we analyze the basic characteristics of the player population, such as the geographic distribution of the players, when and how often they play the game. We also look at the overall NPQ data such as distributions of RTT and capacity.

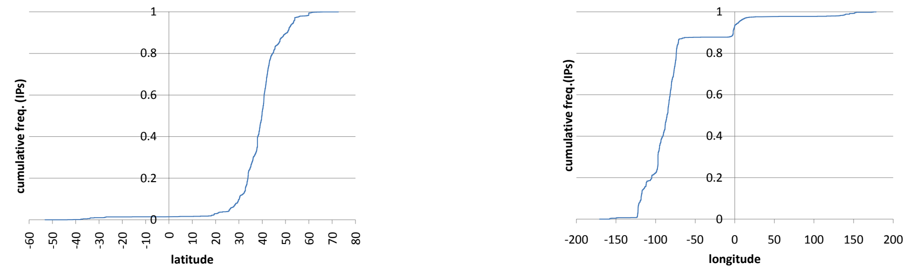

Fig. 3. Latitude and longitude density of players

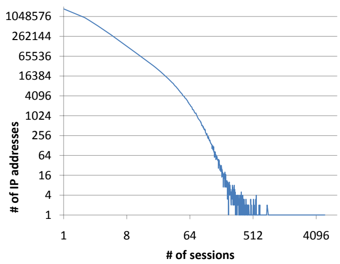

Fig. 4. Game sessions per IP address (log-log)

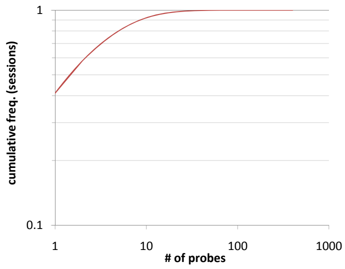

Fig. 5. Probes per session (log-log)

Figure 1 shows the geographic locations of all 5,658,951 unique IP addresses, which correspond to 68,834 unique latitude and longitude coordinates. To examine the density of players in each region, we present Figure 3. Almost 85% of players are in USA - longitudes -130 to -60, and latitudes 30 to 50. Roughly 15% are in western Europe. Since players are spread across this large geographic region, it is quite possible for consoles that are "too far apart" to probe each other. This partly motivates us to consider estimation techniques that will identify such cases before probing.

To see when games were played, Figure 2 plots the number of game sessions in each hour over a representative week. We notice a very strong diurnal pattern with peaks in the evenings in North American time zones - this is not unexpected given the high density of players in USA. We examine game playing behavior in more detail in Figure 4. The number of games attempted from each IP address almost follows a Zipf distribution. In the far right of the graph, one IP address attempted 5438 sessions - over a 50 day period, this is a huge number of games for any 1 individual! We suspect that the IP addresses in this area of the graph are for proxies with many players behind them.

Figure 5 shows a CDF of the number of consoles hosting a game that were probed in each session. While there are many sessions that probed few consoles, there are some that probed as many as 400 consoles. This number depends on how many game hosts the Xbox Live server gives a console requesting a game, which in turn depends on how many consoles are available at the time and the type of game requested.

Now we consider overall NPQ data. Figure 6 shows the CDF of RTT across all probes. Over 25% of the measurements are above 150ms, which is an upper bound for a responsive user experience in typical FPS games [1]. We want to pre-determine in which cases the RTT will be above 150ms and skip probing altogether, thereby

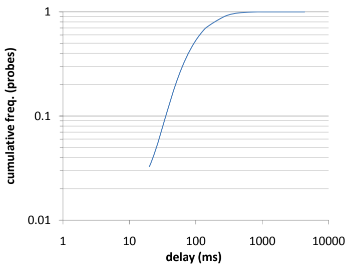

Fig. 6. RTT delay reported by probes (log-log)

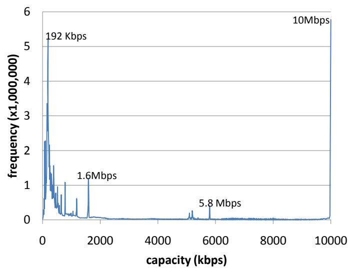

Fig. 7. Downstream capacity reported by probes

potentially reducing the total number of probes by 25%. Figure 7 shows the distribution of measured capacity across all probes, in the direction from the console hosting a game to the console requesting to join it. The graph for upstream capacity is similar. We see peaks around typical capacities for broadband access in USA (e.g. 192Kbps, 1.5Mbps), within some marginal error due to the packet pair estimation technique.

5 NPQ Prediction

The NPQ probing technique that Halo 3 uses consists of 16 packets per console being probed (4 packet pairs in each direction). However, there can be many candidate hosts for a game. For scalability and to minimize user wait time, we want to reduce the total number of probes and hence propose the use of NPQ predictors. Our goal is to estimate apriori if a console has good NPQ to a remote candidate host console, without doing a probe. If bad NPQ is predicted, then this candidate should not be probed. If good NPQ is predicted, then limited probing can be done (e.g. only 1 packet pair). If no prediction can be made, standard probing should ensue. Based on our analysis of the NPQ data, we now propose and evaluate three NPQ predictors.

5.1 IP History Predictor

We hypothesize that a probe between a pair of consoles at time t_1 produces an NPQ estimate that is still valid at a later time t_2. This may be true if the median RTT and average upstream and downstream bottleneck capacities do not vary significantly over a period of $\delta = t_2 - t_1$. To test this hypothesis, and estimate how large δ can be, we examine NPQ data for pairs of IP addresses over different periods of time.

Figure 8 shows the CDF of the coefficient of variation (CV) in RTT for pairs of IP addresses over different time windows. For instance, the “Within 5 min” line shows the CV of RTTs from probes between the same pair of IP addresses within any 5 minute window. To draw meaningful conclusions, we consider only those IP pairs that probed each other at least 5 times during that period. We have plotted similar lines for 30 minutes, 6 hours, 1 day and the entire 50 day trace (the line labeled ”no constraints”). The lines for all 5 time windows overlap with each other. For over 90% of IP address pairs that probed each other at least 5 times, the variation in RTT estimates was minuscule (CV under 0.2), even up to a δ of 50 days. For comparison we plot the “baseline” - instead of considering CV for each pair of IPs, we consider the CV for each single IP.

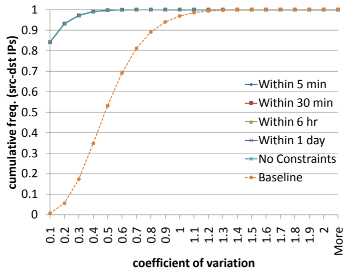

Fig. 8. RTT variation for IP pairs

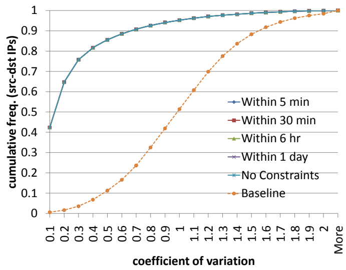

Fig. 9. Capacity variation for IP pairs

That is, for each IP address, the CV across all RTT estimates where this IP address was involved, across the entire trace. This line is well below the others, indicating that the RTTs are spread across a wide range. We conclude that the IP History Predictor can perform quite well for predicting RTT, and the δ can be as large as 50 days.

Figure 9 has the same graph for downstream capacity. The upstream capacity graph is very similar and is omitted for conciseness. Again, δ does not affect the NPQ prediction. While the CV is larger, it is under 0.65 for 90% of the IP pairs, and is still much higher than the "baseline". Thus we believe the IP History Predictor adequately predicts the NPQ between a pair of consoles based on an NPQ estimate from a prior probe.

5.2 Prefix History Predictor

We have shown the IP History Predictor to work only when pairs of consoles have probed each other in the past. This may reduce the number of probes in only a limited set of cases. Thus we now propose the Prefix History Predictor - this is similar to the IP History predictor, except it uses IP *prefix* pairs. We hypothesize that a probe between a pair of consoles A_1 and B_1 at time t_1 produces an NPQ estimate that is still valid at a later time t_2 for a different pair of consoles A_2 and B_2, as long as A_1 and A_2 belong to one BGP prefix, and B_1 and B_2 belong to one BGP prefix.

This predictor may be accurate if consoles in the same prefix share similar last mile access. However, broadband ISPs offer several access packages (e.g., 192Kbps DSL or 1.5Mbps DSL), and the prefix may indicate geographic location more than link speed. Thus, predictions for capacity may be less accurate than for RTT. We now analyze NPQ data for pairs of IP prefixes that probed each other at least 5 times. We find a console's prefix by a longest prefix match against the 12/27/2007 RouteViews BGP table [8].

Figure 10 shows the performance of this predictor for delay, and can be compared to Figure 8. When considering prefix pairs, δ has a bigger impact - the older the original probe, the worse the prediction. Since CV is a relative metric, small variations in small RTTs (e.g. 5ms versus 10ms) can produce a large CV. Thus in Figure 11 we look at the semi-interquartile range (SIQR) of RTT estimates for prefix pairs for no limit on δ (i.e., the "no constraints" case). For 90% of prefix pairs, the SIQR is under 40ms. Thus it is the outliers beyond the 25%-75% SIQR that contribute to this additional variability.

Figure 12 shows the performance of this predictor for downstream capacity estimation. For δ beyond 5 minutes, it is a very poor predictor. We suspect this is due to different subscription levels of last mile capacity within the same prefix.

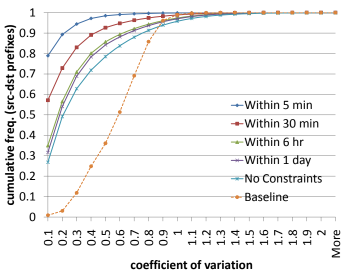

Fig. 10. RTT variation for prefix pairs

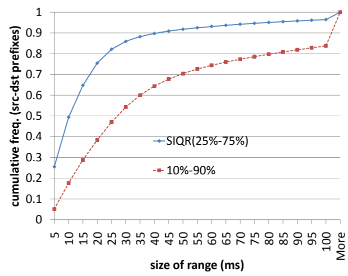

Fig. 11. SIQR of RTT for prefix pairs

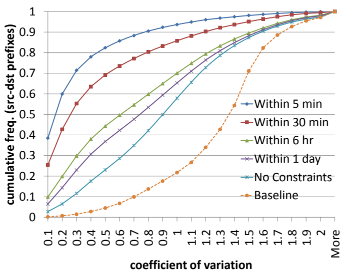

Fig. 12. Capacity variation for prefix pairs

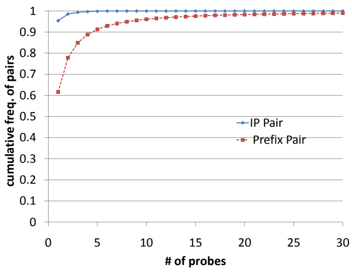

Fig. 13. Number of repetitive probes

Based on these results, we believe the Prefix History Predictor can adequately predict the RTT between a pair of consoles based on an NPQ estimate from a prior probe between consoles in the same pair of prefixes. However, this prediction is not as accurate as the IP History Predictor - so we suggest first applying that predictor, and only if it cannot make a prediction, then using the Prefix History Predictor. To show in how many cases this would apply, we present Figure 13. We plot the CDF of the number of repeated probes in the entire trace between the same pair of IP addresses, and the same pair of prefixes. Only about 5% of pairs of consoles probed each other more than once, while about 39% of prefix pairs probed each other more than once. Note that we have clipped the horizontal axis at 30 for presentation purposes - the IP pair line continues to 114 probes, and the prefix pair line continues to 14,513.

5.3 Geography Predictor

While 39% of prefix pairs is still a significant fraction of the number of consoles, and has the potential to reduce a far larger portion of probes those prefixes probed each other several times, there is still about 61% of prefix pairs left. We now consider the Geography Predictor - we hypothesize that the geographic distance between two consoles has a strong correlation with their RTT, and that current databases for mapping IP addresses to geographic coordinates are reasonably accurate for this. This may be true if distant IP addresses traverse longer links and more router hops to communicate. This predictor does not consider past history, and thus could be applied to any pair of consoles.

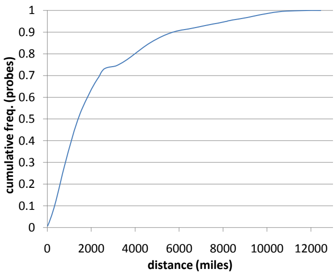

Fig. 14. Probe distance distribution

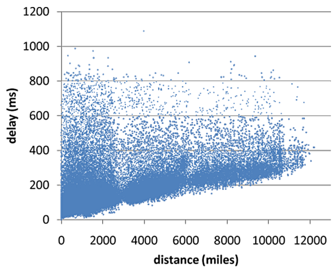

Fig. 15. Distance-RTT correlation

It is not clear why geographic distance would be correlated with down/up stream bottleneck capacity - our analysis indicates the correlations are -0.075 and -0.097, respectively. Thus we omit detailed results on capacity for conciseness and focus on RTT.

We use a MaxMind location database to convert the source and destination IP addresses in a probe to the geographic coordinates, and apply the great circle distance algorithm [9] to calculate distance. The distribution in Figure 14 shows a wide range of distances between probe hosts. About 14% of probes traversed over 5,000 miles, which indicates there is room for optimization by filtering out these console pairs from the probe list. The graph also shows that we have enough samples to examine the correlation between distance and delay.

Figure 15 plots the correlation between distance and RTT for 100,000 randomly selected probes (we were unable to plot over 126 million points on the same graph – different samples of 100,000 points gave us very similar graphs). We see a very strong correlation between the geographic distance and minimum RTT. However, there is a lot of noise above that minimum, which may be due to queuing delays and sub-optimal routes. We conclude that the Geography Predictor is useful for filtering out pairs of IP addresses that are too far apart to have a low RTT.

5.4 Using Predictors in Matchmaking

Incorporating these three predictors into matchmaking is not difficult. For the IP History Predictor, each console will keep a history of previous probes that it was involved in. It can look up this history before attempting any future probes, and decide which candidate game hosts to ignore. For the Prefix History Predictor, the Xbox Live server can filter the set of candidate game hosts it provides to each console based on their prefixes and past probe history. The server already has the past NPQ estimates, and it can easily keep fresh BGP tables to look up prefixes. The Geography Predictor requires an IP to geographic coordinate database, on either the Xbox Live server or on each console.

6 Prior Work

Most prior work on network gaming has focused on games with a client-server architecture [10,4,5] where the server is well-provisioned and dedicated. The literature onP2P games is very limited. In [6], the authors examine game clients deployed in three access networks: dial-up, cable and ADSL. However, their experiments are limited to one client per access network, and use only one cable and one ADSL link. The game traffic of Halo 2 is analyzed in [11] in a LAN environment for traffic modeling and not for end-to-end network characteristics between real Halo 2 players.

There has been much prior work on efficient and accurate NPQ prediction. For conciseness, we identify those done in the context of network gaming. As before, most of this work is for client-server games. In [12], a simple server-side method is proposed to improve server location discovery and reduce probe traffic. Our NPQ prediction methods focus also on reducing overall probe time since that directly affects user wait time. Also, we not only utilize the geographic location of consoles but also previous probe results. A flooding-style server discovery mechanism is proposed in [13] to quickly locate local servers and prevent single directory server failure. That does not scale to P2P games, since in our case several hundreds of thousands of consoles can be on-line at any time. A server selection algorithm is proposed in [14] for distant game players who want to play specifically with each other. Our work considers the general case of joining players to any acceptable game, and thus considers NPQ data and correlators across all on-line consoles. The geographic distribution of game servers and players is used in [15] to redirect players to close game servers. While [16] does not consider on-line games, they correlate geographic location and network delay to find a host, and their experimental result about the correlation complements ours.

Outside network games, there has been a lot of research on characterizing NPQ over the Internet. Many of these [17,18] use PlanetLab nodes. They are mostly located in high-performance and stable academic networks, and thus do not reflect the characteristics of consumer access networks. In [19], the constancy of NPQ over time among 31 hosts is studied within a stable academic network. Our work significantly complements prior work in terms of scale and diversity of network connectivity. Studies of hosts behind consumer broadband lines are rare. It is extremely difficult to build a large testbed of such hosts on the Internet. While [20] characterizes network performance between consumer broadband hosts, they use only 25 hosts. More recently, [21] studies residential network link capacities, RTT, and loss rates through relatively large-scale measurement studies. They use 1,894 hosts behind 11 major DSL and cable providers in North America and Europe. Our study is much larger in scale, involving over 5.6 million hosts. Furthermore, they do not characterize direct network connections between pairs of broadband hosts since they measure from several vantage points located in academic networks. Techniques for estimating NPQ have been studied extensively [7,22]. Our work focuses not on the techniques itself, but on the NPQ data.

7 Conclusions

We studied the quality of network paths among Xbox 360 game consoles playing Halo 3. We focused on network delay and capacity measured between players prior to each Internet game match-up. We studied the general characteristics of the player population such as geographical diversity and diurnal patterns of game play. We leveraged our understanding of these characteristics to propose three predictors for determining path quality without additional probe traffic : IP and prefix history-based and geography-based. Our evaluation of these predictors showed that they can significantly reduce the number of probes and hence user wait time during matchmaking. For future work, we plan on comparing the initial NPQ estimate to actual in-game network performance.

References

1. Dick, M., Wellnitz, O., Wolf, L.: Analysis of factors affecting players performance and perception in multiplayer games. NetGames (2005)
2. Quax, P., Monsieurs, P., Lamotte, W., Vleeschauwer, D.D., Degrande, N.: Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. NetGames (2004)
3. Armitage, G.: An experimental estimation of latency sensitivity in multiplayer Quake 3. In: ICON (2003)
4. Feng, W., Chang, R., Feng, W., Walpole, J.: Provisioning on-line games: A traffic analysis of a busy Counter Strike server. In: IMW (2002)
5. Kim, J., Choi, J., Chang, D., Kwon, T., Choi, Y., Yuk, E.: Traffic characteristics of a massively multi-player online role playing game. NetGames (2005)
6. Jehaes, T., Vleeschauwer, D.D., Coppens, T., Doorselaer, B.V., Deckers, W.N.E., Spruyt, J., Smets, R.: Access network delay in networked games. NetGames (2003)
7. Carter, R.L., Crovella, M.E.: Measuring bottleneck link speed in packet-switched networks. Technical report, Boston University (March 1996)
8. University of Oregon: Routeviews project page `http://www.routeviews.org/`
9. Hexa software development center: Distance calculation method between two latitude and longitude coordinates, `http://zipcodeworld.com/docs/distance.pdf`
10. Chambers, C., Feng, W., Sahu, S., Saha, D.: Measurement-based characterization of a collection of on-line games. In: IMC (2005)
11. Zander, S., Armitage, G.: A traffic model for the Xbox game Halo2. In: NOSSDAV (2005)
12. Zander, S., Kennedy, D., Armitage, G.: Server-discovery traffic patterns generated by multiplayer first person shooter games. NetGames (2005)
13. Henderson, T.: Observations on game server discovery mechanisms. NetGames (2003)
14. Gargolinski, S., Pierre, S., Claypool, M.: Game server selection for multiple players. NetGames (2005)
15. Chamber, C., Feng, W., Feng, W., Saha, D.: A geographic redirection service for on-line games. ACM Multimedia, New York (2003)
16. Fdida, S., Duarte, O.C.M.B., de Rezende, J.F., Ziviani, A.: Toward a Measurement-Based Geographic Location Service. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 43–52. Springer, Heidelberg (2004)
17. Lee, S.-J., Basu, S., Sharma, P., Banerjee, S., Fonseca, R.: Measuring Bandwidth Between PlanetLab Nodes. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 292–305. Springer, Heidelberg (2005)
18. Banerjee, S., Griffin, T.G., Pias, M.: The Interdomain Connectivity of PlanetLab Nodes. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 73–82. Springer, Heidelberg (2004)
19. Zhang, Y., Duffield, N., Paxson, V., Shenker, S.: On the constancy of Internet path properties. In: IMW (2001)
20. Lakshminarayanan, J., Padmanabhan, V.N.: Some findings on the network performance of broadband hosts. In: IMC (2003)
21. Dischinger, M., Haeberlen, A., Gummadi, K.P., Saroiu, S.: Characterizing residential broadband networks. In: IMC (2007)
22. Dovrolis, C., Ramanathan, P., Moore, D.: Packet-dispersion techniques and a capacity-estimation methodology. IEEE/ACM Transactions on Networking (December 2004)