Network Characteristics for Server Selection in Online Games

Mark Claypool
CS Department at Worcester Polytechnic Institute, Worcester, MA, USA

ABSTRACT

Online gameplay is impacted by the network characteristics of players connected to the same server. Unfortunately, the network characteristics of online game servers are not well-understood, particularly for groups that wish to play together on the same server. As a step towards a remedy, this paper presents analysis of an extensive set of measurements of game servers on the Internet. Over the course of many months, actual Internet game servers were queried simultaneously by twenty-five emulated game clients, with both servers and clients spread out on the Internet. The data provides statistics on the uptime and populations of game servers over a month long period an an in-depth look at the suitability for game servers for multi-player server selection, concentrating on characteristics critical to playability – *latency* and *fairness*. Analysis finds most game servers have latencies suitable for third-person and omnipresent games, such as real-time strategy, sports and role-playing games, providing numerous server choices for game players. However, far fewer game servers have the low latencies required for first-person games, such as shooters or race games. In all cases, groups that wish to play together have a greatly reduced set of servers from which to choose because of inherent unfairness in server latencies and server selection is particularly limited as the group size increases. These results hold across different game types and even across different generations of games. The data should be useful for game developers and network researchers that seek to improve game server selection, whether for single or multiple players.

1. INTRODUCTION

The growth in capability and penetration of broadband access networks to the home has fueled the growth of online games played over the Internet. As this article is being written, it is 4am (EST) on a typical weekday morning and Gamespy Arcade* reports more than 310,000 players online playing over 100,000 games! This proliferation in online game players has been matched by an equivalent growth in the variety of online game offerings. The spectrum of online games has shifted from the 1990's where a few players collaborated or competed on a Local Area Network (LAN) in first-person perspective games such as id's *Doom*, to thousands of players interacting over the Internet in a wide variety of games ranging from first-person shooter games and role playing games to real-time strategy and sports games. This escalation in the popularity of online games is also reflected in the correspondingly high number of game servers spread across the globe that support thousands of these players.

Most online games deploy a client-server model of interaction and so could seemingly benefit from network enhancements that support traditional client-server applications. Unfortunately, online games often do not benefit because of significant architectural differences. Traditional clients need specific content from a server and are not interested in alternate versions in the content. In contrast, game clients can often connect to a variety of servers for a different, yet acceptable, gameplay experience with alternate content (maps or players). Traditional applications care mostly about network throughput, while online games are care mostly about network latency. Traditional applications can benefit from transparent caching and local content access, such as content distribution networks and increased bandwidths. On the other hand, online game players want control over their server selection and especially need specific servers when playing online simultaneously with friends or family.

Many online games allow players to choose from among many servers for online play. For many games, this arrises because users can run their own game servers, allowing clients to connect from anywhere on the Internet. Nearly all popular first-person shooter games (such as *Quake*, *Doom*, and *Unreal Tournament*) allow users to run

Further author information: email - claypool@cs.wpi.edu, phone - 1 508 831 5409

*Gamespy provides popular services for the online gaming community, including game server browsing and player forums. Gamespy Arcade is online at http://www.gamespyarcade.com/.

their own game servers. Similarly, most real-time strategy games (such as *Warcraft* and *Age of Empires*) allow users to host a game, thus providing many server choices for clients playing online.

And the choice of game server impacts the online experience. Game servers can reach maximum player capacity, require clients to install cheat protection software (such as *PunkBuster*), or limit access to clients to specific versions or mods. The game map, game configuration and other in-game parameters (such as having friendly-fire enabled for a team-based first-person shooter) can influence a player's choice to join a particular game server.

Even if all preferential game conditions are met by a game server, network latency will also impact the gameplay experienced. The range of latencies from a game client to all available game servers can be as broad as the range of end-to-end Internet latencies, going from milliseconds for a local game to thousands of milliseconds for a game across the world or over a congested, limited bandwidth network. Previous work has demonstrated that latencies can degrade player performance,[1–7] making selection of a fast server important for good online gameplay.

The problem of game server selection can be compounded when multiple players want to play online simultaneously on the same game server. This can arise when friends and family arrange for an on-line gaming session or when more formalized teams of players (typically called *clans*) compete against other teams during a pre-arranged match. Given the increasingly connected nature of the world, such players are increasingly likely to be geographically separated. A server that is fast for one player may be slow for the next player and vice versa. Support for finding a game server that performs acceptably for all players that want to play together remains an open research issue.

In order to improve game server selection, both for single and multiple players and for the increasingly diverse set of online games they play, there is a need for a better understanding of the network characteristics of current game servers. Understanding the availability and performance for existing game servers will provide insights as to whether there need to be alternate means for server deployment. Measuring game server performance from game clients simultaneously running on different Internet nodes will enable assessment of support for both single and multiple players, and can be extrapolated to assess support for a wide-range of game genres.

Some related work[8–10] has looked at improving server selection for the current client-server online games, but has focused almost exclusively on helping single players without considering support for server selection for simultaneous players. Other related work[11,12] proposes alternate structures for organizing game servers to provide for better online gameplay, but are not immediately useful for today's predominantly client-server online game environments. Related work in third party architectures[13,14] shows promise in improving server selection for games and other client-server applications, but does not work without explicit buy-in by game developers. Other research has analyzed game traffic from the point of view of a server[15,16] or a client,[17–20] but does not specifically consider traffic patterns or performance criteria for a group of servers and a group of clients.

In an effort to provide a better understanding of current game server selection, this paper provides a network characterization of actual game servers on the Internet. An extensive set of measurements were gathered over the course of several months, involving 25 geographically dispersed game clients and 60 geographically dispersed game servers from 3 different game types. The actual Internet game servers were queried simultaneously by the game clients, providing statistics on the uptime and populations of game servers and an in-depth look at the suitability for game servers to support single and multiple player server selection for a range of game genres.

Analysis of the data finds most game servers have latencies suitable for third-person and omnipresent games, such as real-time strategy, sports and role-playing games, providing numerous server choices for game players, whether selecting a server alone or simultaneously with friends and family. However, far fewer game servers have the low latencies required for first-person games, such as shooters or race games. Groups that wish to play together have a reduced set of servers from which to choose because of inherent unfairness in server latencies. These results hold across different game types and even across different generations of games.

The rest of this paper is organized as follows: Section 2 provides background on game server browsing; Section 3 describes our methodology to measure the network characteristics for online games; Section 4 analyzes the results; Section 5 summarizes our conclusions and presents possible future work.

2. GAME SERVER BROWSING

Game server browsing allows players to find and select an online game server for their game clients. For online games that allow users to host their own game servers, game server browsing proceeds as follows.

A master server for a specific game, typically hosted and run by the company that created the game, runs at a well-known Internet domain name and port. This name and port number are hard-coded into the game clients or can be looked up by the player via an Internet search.

Upon starting, a game server registers with their master server, providing system and game information. The master server keeps this information in its game server directory until the game server gracefully shuts down and unregisters, or until the master server purges the information when it has not received a periodic update from the individual game server.

A game client connects to their master server to obtain information on the individual game servers available at the moment. The master server provides information on the IP address for each game server node, as well as the port number on which the game server is listening.

The game client then has the option to individually query each server returned in the list from the master server, in order to verify that each server is, indeed, up and to update statistics on the number of players, map type, etc. These individual queries also provide an estimate of the latency from the client to the server, measured as the time elapsed between sending out the single-packet query until receiving a single-packet response. Typical game clients (and players) call this estimate the "ping" time, but it is different than ICMP ping times since game ping times are application-layer to application-layer while ICMP packets are handled at the operating system layer and are sometimes treated differently by network routers. Typically, game ping times are slightly (about 10 milliseconds) higher than are ICMP ping times along the same network path.[21] While there are other measures of performance that may affect online gameplay, such as packet loss and available bandwidth, player performance is typically dominated by network latency (also called "lag" by game players).

Figure 1 depicts a screen shot of a typical in-game server browser, in this case from the game *Counter-Strike: Source*. The main window shows the list of possible game servers, indicated with a text string name created by the user that is hosting the game. Information about the number of players, the map being played and the latency (in milliseconds) is provided for each game server. On the far left, the lock symbol indicates whether the game server is password protected or open to general access. At the bottom, a simple interface allows the user to apply queries to filter the list of results based on map, latency, population or other preferences. The tabs include an option to browse Internet-wide games or only LAN games.

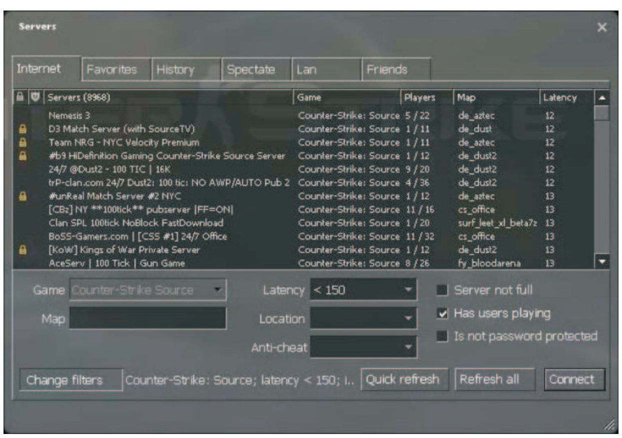

Figure 1. Screen Shot of Typical In-Game Server Browser

Figure 2. Geographic Location of Game Clients and Game Servers Used in this Study.

3. METHODOLOGY

3.1. Game Selection

While there are potentially numerous commercial game types to study, most use the same model for server browsing described in Section 2. Thus, the decision was made to concentrate on games produced by one commercial vendor with the assumption that the observed characteristics would generally hold for game servers from other vendors.

The vendor chosen was *id Software*[†], as they created the game *Doom* (in 1993) that brought a deluge of related games that today make up the bulk of game-types available online through server browsing.[‡] Since the creation of Doom, id has gone on to create *Doom 2* and *Doom 3* and the popular Quake series (through to *Quake IV*). The games Doom 3 and Quake IV (henceforth called *Quake 4*) were selected as representative examples of current games (released in 2004 and 2005, respectively), allowing comparison with each other. Quake 3 (released in 1999) was selected as a representative example of a previous generation game, but one that is still quite popular, allowing comparison across game generations.

3.2. Master Servers

Data was gathered from the master server for each of the selected games for a one month period. The purpose of this was two-fold: 1) long-term analysis of data from the master server can provide insights into the general ebb and flow of servers and player populations on those servers; 2) game servers that are consistently up (we call these *permanent* servers) were used in the second phase of data gathering, allowing individual clients to reliably obtain server data over a long period of time.

The well-known domain names for the master servers for the selected games are `master3.idsoftware.com`, `q4master.idsoftware.com`, and `idnet.ua-corp.com` for Quake 3, Quake 4 and Doom 3, respectively. For master server browsing and analysis, the location of the client is not relevant as the same information is returned by the master server regardless of the client location.

Master server data was gathered for the month of December 2006. From analysis on the individual game servers (see Section 4 for more details), 20 permanent servers for each game (Quake 3, Quake 4 and Doom 3) were selected, depicted geographically in Figure 2. These permanent servers were used for queries by individual clients during the second phase of the study. The IP addresses and port numbers for the servers can be found online.[§]

3.3. Emulating Simultaneous Game Clients

The intent was to gather game server information, as would be seen by a typical game server browser in a game client, from actual Internet clients to actual Internet game servers. This provides insight into server characteristics for a single player seeking an online game. In addition, the intent was to gather information from multiple game clients *simultaneously* in order to represent multiple players looking for a suitable game server they could play on together. This provides insight into the problem space for game server selection for players that wish to play together as a group.

Qstat. In order to emulate actual game clients, *Qstat*[¶] was used. Qstat is an open source game server browser that provides server information for a variety of game types. This is particularly useful since the information can be obtained from commercial, closed-source games without requiring installation of commercial game clients on each node. Qstat can be run from the command line, making it ideal for deployment in an automated fashion on remote computers, such as during our experiments.

Qstat includes support for both older and newer games, including all versions of *Quake* and *Hexen*, all versions of *Unreal Tournament*, versions of *Half-Life* and *Counter-strike*, versions of *Tribes* and numerous other games. The games selected for this study, *Quake 3*, *Quake 4*, and *Doom 3*, are supported.

[†] http://www.idsoftware.com/

[‡] In fact, Doom was so popular that first-person shooter games were called "Doom clones" until the late 1990s.

[§] http://www.cs.wpi.edu/~claypool/papers/game-server/

[¶] http://www.qstat.org/

Qstat can be used to query both master servers and individual game servers, providing information on the: name, IP address and latency of the server; maximum and current number of connected players; latency and scores for connected players; and the map and rules specific to the server. Qstat can produce results on screen or written to a file in text, HTML or XML. Our study used XML to make use of the variety of XML tools to ease parsing of data during the analysis.

Game Clients. In order to use actual Internet nodes, PlanetLab[‖] was used to provide a platform for the game clients from different Internet locations. Twenty-five clients were selected, geographically spread out on the Internet to reflect the truly global pervasiveness of online gaming, depicted geographically in Figure 2. Clients were selected from Australia, France, Germany, Israel, Japan, Korea, Puerto Rico, Spain, the United Kingdom, and the United States, The domain names for the clients can be found online.[**]

Each PlanetLab client ran a custom `perl` script called the *Gatherer* that took as input the game type and a list of the individual game servers to query. The Gatherer then iterated through the server list, launching Qstat with the appropriate parameters, writing the output in XML to data files organized by client, server, game type, date and time.

Organizer. The launching of the emulated game clients on the PlanetLab nodes was coordinated from a custom `perl` and `expect` script called the *Organizer*. The Organizer first formed `ssh` connections to each of the clients, then simultaneously instructed them to launch their Gatherer scripts. Then, the Organizer monitored each client for completion of the Gatherers, whereupon the clients were instructed to send data back to the Organizer via `scp`.

The Organizer itself was controlled by means of a `cron` job, run from a Linux PC at Worcester Polytechnic Institute in Massachusetts, USA.

3.4. Game Servers

The Organizer was launched every half-hour. Data collection took place for a one-week period shortly after the data on the master servers was gathered and analyzed (Section 3.1). Altogether, this resulted in over 500,000 queries from individual game clients to individual game servers, and over 1000 comparisons for simultaneous players to the same game servers.

4. ANALYSIS

Analysis proceeds first with the data from the master server for each game type providing game server characteristics (Section 4.1), then with the distributed client queries to the individual game servers to illustrate game server selection (Section 4.2)).

4.1. Server Characteristics

4.1.1. Number of Servers

Analysis using the month-long master server data provides a measure of the number of servers for each game type, shown in Table 1. Despite being older, Quake 3 has more servers than does Quake 4 with Doom 3 having by far the fewest. While not all of the servers returned by the master server are actually up (responding to Qstat queries), about 90% of servers listed are reachable.

Analysis of the up servers over the month provides a means to visualize any correlation in number of game servers to day of the week. Figure 3 depicts the results, with the x-axis being the day of the week and the y-axis the number of servers obtained during each half-hour query. The number of servers is relatively constant across the month, with a slight upward trend in Quake 4 servers and a slight downward trend in Doom 3 servers. More importantly, there is no strong visual correlation with number of servers and day of the week.

Figure 4 depicts a zoomed in version of Figure 3, showing the number of servers over the course of one day. Here, one game type (Doom 3) is depicted in order to allow data focus, but the other games (Quake 3 and Quake

[‖] http://www.planet-lab.org/
[**] http://www.cs.wpi.edu/~claypool/papers/game-server/

Game	Listed Mean	Up Mean	Up Std Dev	Up CoV
Quake 3	1642	1397 (85%)	52.6	0.04
Quake 4	553	521 (94%)	15.9	0.03
Doom 3	76	67 (88%)	8.5	0.13

Table 1. Aggregate Statistics for Game Servers for One Month

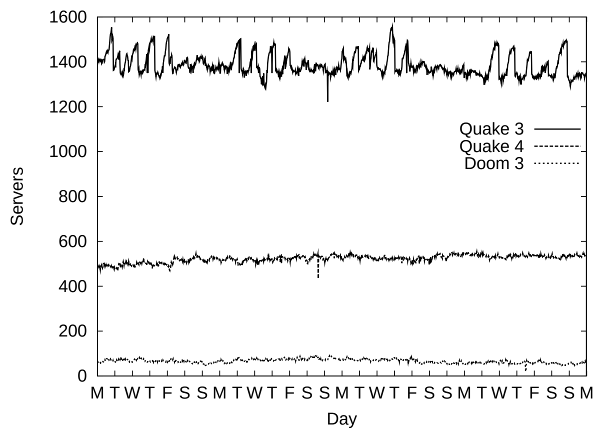

Figure 3. Number of Servers Up for All Games for One Month

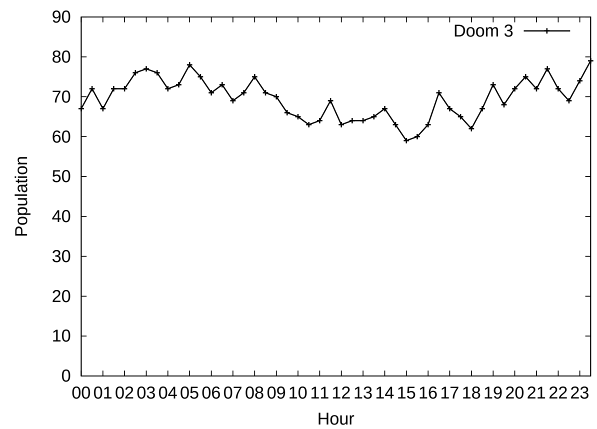

Figure 4. Number of Servers Up for One Game (Doom 3) in One Day

4) show similar trends. Figure 4 has no clear visual correlation with the number of servers and the time of day.[††] This is perhaps explained by the global nature of online gameplay, where it is alway peak activity time (typically late afternoon to early evening) at some place in the world.

4.1.2. Number of Players

Analysis of the total number of players across all game servers for each game type is provided in Table 2. Similar to the trends in number of servers, Quake 3 has, on average, the most players with Quake 4 and Doom 3 trailing.

Game	Mean	Std Dev	CoV
Quake 3	2641	619	0.23
Quake 4	247	102	0.41
Doom 3	71	37	0.52

Table 2. Aggregate Statistics for Total Player Population for One Month

Using the data from Table 1 and Table 2, there are on average 1.3 players for every Quake 3 server, 0.45 players for every Quake 4 server, and 0.93 players for every Doom 3 server. However, the population spread of players among servers is not uniform, with some servers being filled more than others. Figure 5 depicts the cumulative distribution of the number of players on a server divided by the player capacity on the server (converted to a percentage). For all game types, half or slightly over half of the servers are completely empty of players. Also, there are few servers that are completely full for any game type, which is fortunate for players looking for a server that is not full.

The Coefficient of Variation (CoV) for the number of players (in Table 2) is much higher than the CoV for the number of servers (in Table 1), suggesting the players come and go more often than the servers. Figure 6

[††]Times are all U.S. Eastern Standard Time, EST.

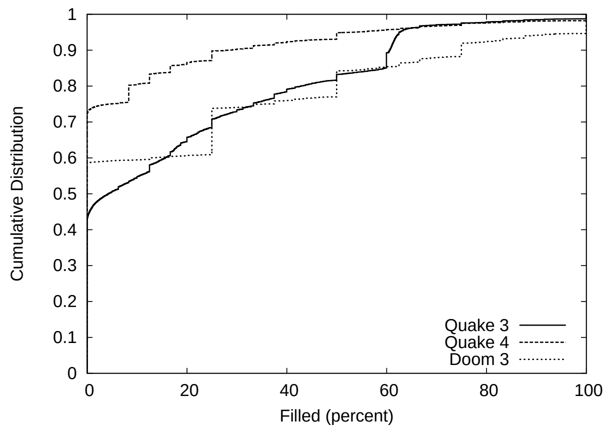

Figure 5. Cumulative Distribution of Percentage Filled

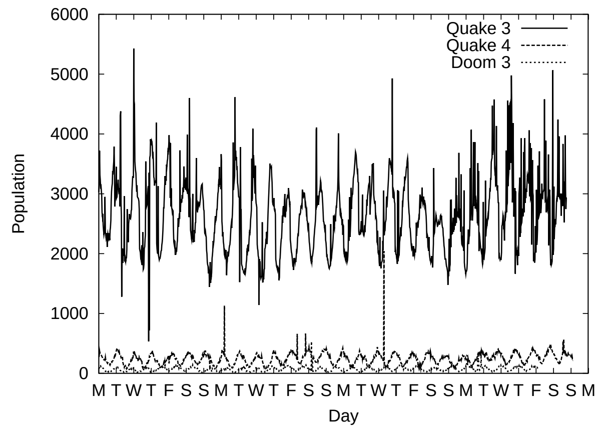

Figure 6. Total Player Population for All Servers for All Games for One Month

depicts the player population over the month for all games, and Figure 7 depicts the player population over a single game (Doom 3 – again, the other games were similar). Visually, there is considerably more variation in the data lines in Figure 6 than in Figure 3, but still no visual correlation between player population and day of the week. Unlike Figure 4, Figure 7, does show a correlation for the total number of players with time of day, with a rise in servers in the late morning (EST), peaking mid-afternoon, and declining until early evening. This is somewhat contrary to the analysis for number of servers, suggesting game players are not so evenly distributed around the world such that they exhibit time of day correlations in their game activity when seen in aggregate.

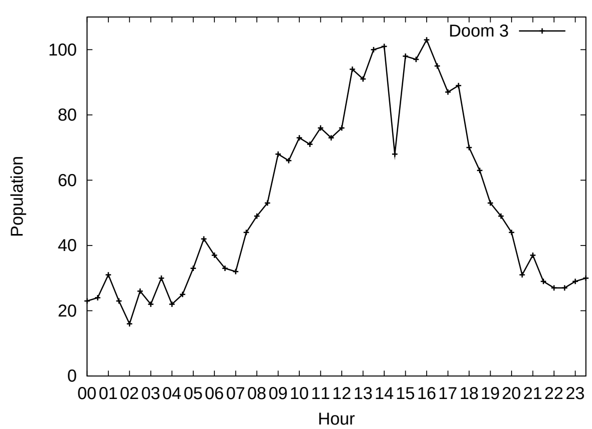

Figure 7. Total Player Population for All Servers for One Game (Doom 3) in One Day

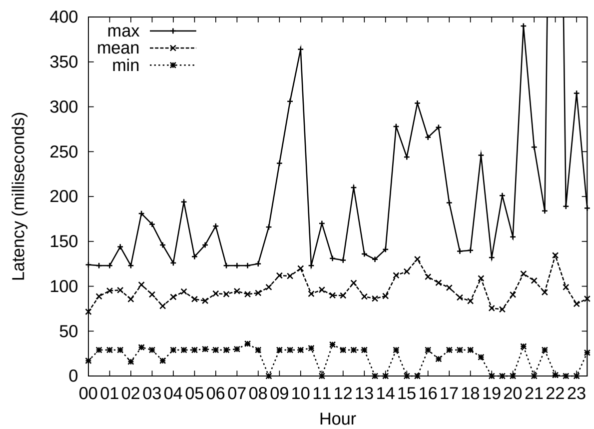

Figure 8. Latency for Single Client for One Game (Doom 3) in One Day

Given the visual time of day correlation with the number of players, it is natural to investigate if there is a visual correlation with network latency to the servers. Figure 8 depicts the network latencies for a single Doom 3 client over the course of a day. Each query (run every half-hour) returns a list of possible Doom 3 servers available for play. Analysis of this list provides a minimum, average, and maximum latency, shown as trendlines in Figure 8 and provided in summary statistics in Table 3. While the trend lines do vary in Figure 8, particularly for the minimum and maximum latencies that have a modest CoV, there is not the same strong visual correlation with time of day as there is with player population in Figure 7.

Given the lack of visual correlation for time of day and network performance (latency to servers) and server populations, and the lack of visual correlation for day of the week and player and server populations, for all

Statistic	Mean	Std Dev	CoV
Min	20.5	13.1	0.64
Mean	95.5	13.6	0.14
Max	205.3	151.9	0.74

Table 3. Aggregate Latency Statistics (in milliseconds) for Single Client for One Game (Doom 3) in One Day

subsequent analysis, the data sets for every day across the entire month are combined.

While Table 1 provides statistics on the percentage of servers returned by the master server that are up (about 90%), it does not indicate if these are the *same* servers that are up each half-hour query or if it is an entirely different set of servers. Over the course of the month, the fraction of times each server (identified by IP address and port number) appears in a master server list over the course of the month is computed. Figure 9 depicts the results, showing a cumulative distribution of up fraction for each game type. The trends for each game are remarkably similar, with all game types generally having a bi-modal distribution. About 80% of servers are up only a handful of times and then are not up for the rest of the month, while about 5% of the servers are up for nearly the entire month. Most likely, the 80% group of servers are launched by individual players for the duration of their game session, with their next game session launching a different server. We call these servers that come and go *ephemeral* servers. The 5% group of servers are dedicated in that they stay up beyond the length of the game session of the player that launched them. We call these *permanent* servers.

For the purposes of the second phase of data collection, where many individual clients query specific servers, the 20 specific servers for each game were selected from the group of permanent servers. The IP addresses and port numbers for the servers can be found online.[‡‡]

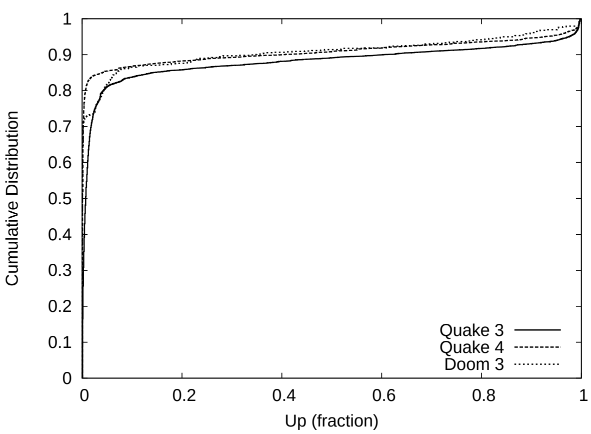

Figure 9. Cumulative Distribution of Server Up Fraction for One Month

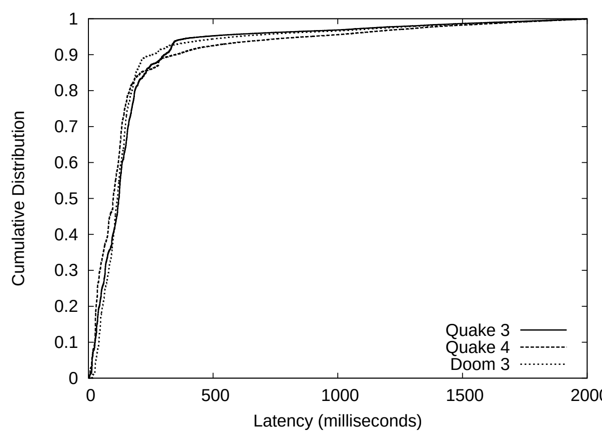

Figure 10. Cumulative Distribution of Latencies for One Month

The last aspect of master server data analysis examines the latencies reported by all game servers over the course of the month in order to see if there is a difference among the different game types. Figure 10 depicts a cumulative distribution of latency for each game type. There is an approximate uniform distribution of latencies from 0 to 400 milliseconds, whereupon the distribution has a heavier tail up to 2000 milliseconds. QStat has a timeout value of 2000 milliseconds, and while theoretically there could be latency values that are higher, such high latencies are above tolerable limits for nearly all online game players. Most importantly, however, note that the distributions for the three games are nearly identical. This suggests there is no quantitative performance difference among the game servers in terms of network latency. Thus, for analysis for individual clients in the next section (Section 4.2), the data sets for the three game types are combined.

[‡‡] http://www.cs.wpi.edu/~claypool/papers/game-server/

4.2. Server Selection

4.2.1. Latency and Player Performance

In order to analyze the general ability of the pool of available game servers to support online gameplay, it is important to first understand the effects of network latency on player performance.

Earlier work[22] provided a classification for online games based on their perspective (*first-person* or *third-person*) and on their model (*avatar* or *omnipresent*). Practically, all games fall into one of three categories: first-person avatar (ex: first-person shooter, racing), third-person avatar (role-playing, side-scrolling) or omnipresent (real-time strategy). A meta analysis of previous work that measured the effects of latency and online games, normalized and combined, provides a way to quantify the effects of latency on online games.

Figure 11 summarizes the meta data analysis of performance versus latency for the different classes of online games, depicted by an exponential curve fit to the previously measured data. The horizontal gray area around 0.75 in Figure 11 is a visual indicator of typical player tolerances for latency. The exact threshold depends upon the game (and to some extent, the player), but generally game performance above this threshold is acceptable while game performance below this threshold is unacceptable. Overall, games that use the avatar model of player interaction are more sensitive to latency than games that use the omnipresent model, and games that use the first-person perspective are more sensitive to latency than games that use the third-person perspective.

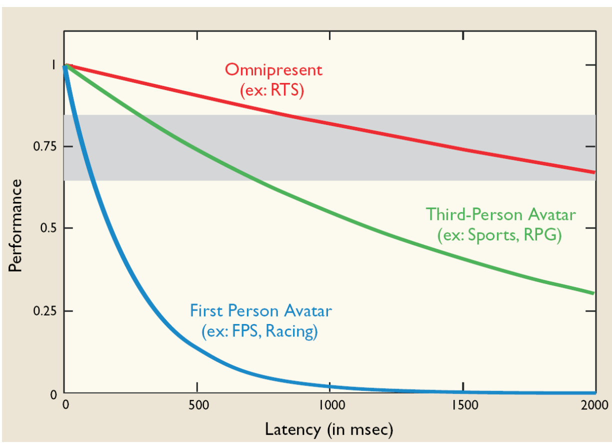

Figure 11. User Performance Under Different Induced Latencies for Several Classes of Games

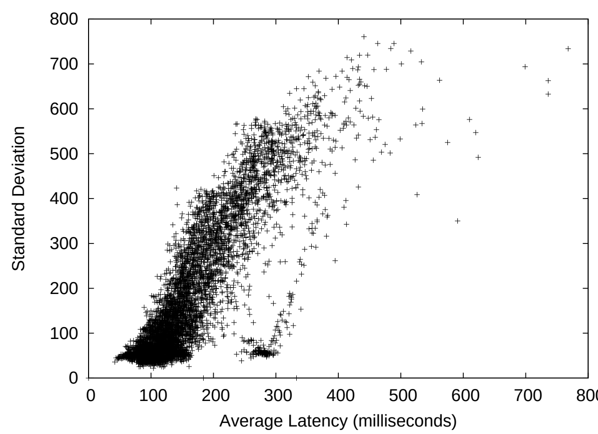

Figure 12. Scatter Plot Unfairness (Standard Deviation) versus Performance (Average Latency) for Twenty Simultaneous Game Players

Table 4 provides a summary of the effects of latency on player performance for the three classes of games. The "Thresholds" provided in the table serve as a yardstick with which to analyze the current ability of game server pools to support a range of online games.

Model	Perspective	Example Genres	Thresholds
Avatar	1st Person	FPS, Racing	100 msec
	3rd Person	Sports, RPG	500 msec
Omnipresent	Varies	RTS, Sim	1000 msec

Table 4. Summary of the Effects of Latency and Online Games

4.2.2. Server Suitability

For single players seeking a server with latency suitable for their online game play, in the absence of other, external criteria (such as map preference), the server with the lowest latency will be the best.

However, for multiple players that wish to play together on the same server, the shared latency of the server becomes important. A natural measure of how "good" a server is for two or more players is to compute the average latency for all players. However, the differences in the latency can have an affect on performance, too. When one player is further away (has a higher latency) from the server than another player, there may be unfairness in the gameplay. For example, suppose two players defeat a monster after a battle. The server, controlling the game world state, generates some treasure as a reward for the players to pick-up. A packet having the location of the treasure is simultaneously sent to each of the players. If one player has low latency, that player can act first and gather the loot. If the other player has high latency, that player will respond more slowly, probably after the first player has gathered the treasure. This unfairness can degrade the gameplay for many games.

A given server can be placed into a two-dimensional space with one axis representing the average network distance (computed as the average latency for all players) and the other axis representing the (un)fairness (computed as the standard deviation of latency across all players). Figure 12 depicts a scatter plot of unfairness versus distance (standard deviation versus average latency) for the entire population of game servers as seen from 20 simultaneous clients. In general, as the average latency increases, the unfairness (standard deviation) of the server also increases. Moreover, this unfairness increases at a faster rate, with a 2x increase in latency resulting in a 2.5x increase in unfairness.

While Figure 12 is visually interesting, using standard deviation as a measure of unfairness does not provide an easy way to map this to player tolerance for acceptable gameplay. However, latency compensation techniques deployed by many game servers do provide a means to quantify the effects of disparate latencies. Notably, a commonly deployed means of reducing the impact of latency is to manipulate game time.[23] With *time delay*, messages arriving at the server are delayed until the message from the client with the highest latency has arrived. Correspondingly, messages from the server are sent out first to the client that is farthest away, delaying messages to closer clients so that they have the same effective latency. While making latencies fair among clients, time delay makes the effective latency the same as the client with the maximum latency. Thus, the maximum latency for any player provides an approximation of how suitable a given server is for simultaneously connected players.

Figure 13 depicts the cumulative distribution of maximum latencies for players simultaneously browsing for a game server, in groups of 2, 5, 10 and 20 simultaneous players. For comparison, the latencies available to the same players choosing a server independent of the other players is also depicted. Looking at how the trendlines shift from left to right, there is a clear increase in the maximum latency as the number of simultaneous players increases. In addition, all the trendlines have a "knee" where the bulk of the latency values lie below this point and the rest of the latency values lie above this point. For example, for two simultaneous players, about 80% of the servers have a maximum latency (for one of the players) of 250 milliseconds or less. However, this knee-effect is less pronounced for higher numbers of simultaneous players.

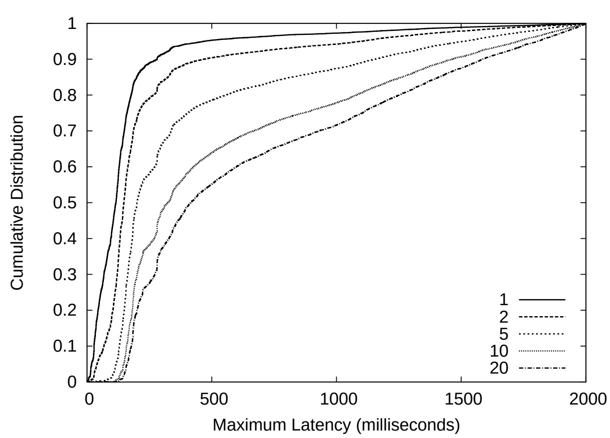

Figure 13. Cumulative Distribution of Maximum Ping for Different Numbers of Clients Playing Together

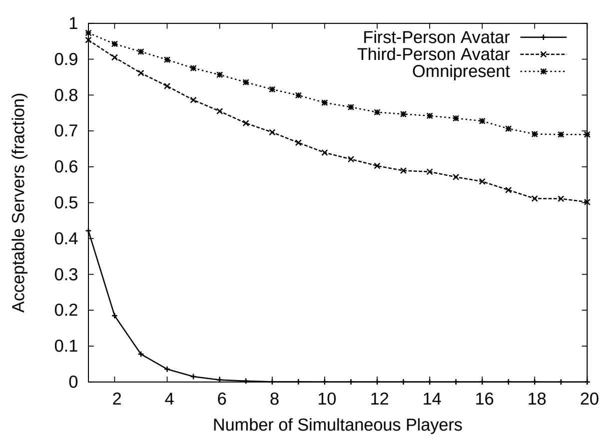

Figure 14. Fraction of Acceptable Servers versus Number of Clients Playing Together for Different Classes of Games

Using the thresholds on acceptable performance from Table 4, the data in Figure 13 can be used to determine the ability of current game servers to support different types of games for simultaneous players. To help visualize, vertical lines can be drawn at the tolerance thresholds for different classes of games: 100 milliseconds for first-person avatar, 500 milliseconds for third-person avatar, and 1000 milliseconds for omnipresent. The distribution points to the left of these lines have acceptable performance for the corresponding game class, while the distribution points to the right have unacceptable performance.

Figure 14 depicts the fraction of acceptable servers versus the number of players that wish to play together simultaneously on the same game server. The three trendlines represent first-person avatar games (such as a first-person shooter), third-person avatar games (such as a role-playing game) and omnipresent games (such as a real-time strategy game). At the far left is the fraction of servers that are acceptable for a single player game. In general, game servers are well-suited to support both third-person avatar and omnipresent games. Single players find over 95% of the servers acceptable, while even up to 20 players seeking a server as a group will find over half the servers suitable. However, for first-person avatar games, much more demanding in terms of latency than other game types, the story is much different. First-person avatar games can generally support single player games, although slightly less than half the servers are acceptable. However, as the number of simultaneous players increases, the fraction of acceptable servers quickly decreases, with almost no servers being acceptable for 7 or more players trying to play together as a group.

5. CONCLUSIONS

The increasing growth in computer networks has brought with it an increase in the interest and importance of online games. Most commercial games use a client-server architecture and many allow the players to connect their clients to their choice of game server. Given that players can also deploy their own server, many games provide players with a choice of many possible game servers. And server selection matters, not only for physical parameters such as player population and game map, but because the latency between the game client and the server has been shown to degrade gameplay. The selection process is made more difficult when players want to play together as a group, co-located on the same game server.

In order to enhance server browsing to support simultaneous players and consider alterations to support the increasingly wide-range of online games, there is first a need for a better understanding of the network characteristics of current game server browsing. This paper provides this needed first step by gathering data on real game servers and clients on the Internet. Months of master server data for three different games provide a characterization of server uptimes and populations, allowing observation of time of day and day of week correlations. A week of game server data gathered from custom software that emulates the server browsing of players seeking to play simultaneously on the same game server provides insight into the ability of currently deployed game servers to support online gameplay.

The results allow us to draw the following conclusions: **1)** There is no visual day of week correlation to server uptime or player population. **2)** There is no visual time of day correlation to server uptime, but there is some correlation with player population. However, there is not a corresponding correlation with server performance (latency). **3)** Game server performance (latency) is nearly independent of game type and game generation. **4)** The number of simultaneous players in a group directly reduces the performance for all players by increasing the maximum latency. **5)** Game server pools are well-suited to support typical third-person games, such as role-playing games or real-time strategy games. The pool of available game servers for third-person games can fairly easily support up to 20 simultaneous game players. **6)** Game server pools are not as able to support typical first-person games, such as first-person shooters or racing games. Players selecting a server outside of a group can find an adequate number of suitable servers, but the pool of servers that provide acceptable performance decreases rapidly with an increase in the group size.

Since the data obtained in this study has been made available to the public, additional processing of the data may provide other insights into game server browsing: 1) Game servers provide information on the latencies and scores of players currently connected to the server. This data can be analyzed to study the range of latencies currently in use, and perhaps correlated with user scores. Or, 2) Geographic information may play a role in the ease (or difficulty) in simultaneous users finding a suitable game server. Additional analysis could examine the physical relationship among the clients and servers, both geographically (in terms of physical distance) and topologically (in terms of network distance), to better understand server browsing.

Some games have servers that are not setup or controlled by individual users, such as servers for one of the popular massively multi-player online (MMO) games. These servers typically have similar server selection issues and so may benefit from the analysis in this paper, but often have the selection done implicitly with a single head node re-directing players to appropriate servers. Study of server selection in this process, probably with support from industry, may be an interesting area of future work.

Acknowledgments: I would like to thank Yong Heng Wu Fang and Nathan T. Tibbetts for their work in gathering the experimental data. This research was partially supported by NSF grant CNS-0423362 for purchase of the Planet Lab nodes at WPI.

REFERENCES

1. G. Armitage, "An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3," in *Proceedings of the 11th IEEE International Conference on Networks (ICON)*, (Sydney, Australia), Sept. 2003.
2. T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool, "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003," in *Proceedings of ACM NetGames*, (Portand, OG, USA), Sept. 2004.
3. J. Nichols and M. Claypool, "The Effects of Latency on Online Madden NFL Football," in *Proceedings of NOSSDAV*, (Kinsale, County Cork, Ireland), June 2004.
4. L. Pantel and L. C. Wolf, "On the Impact of Delay on Real-Time Multiplayer Games," in *Proceedings of NOSSDAV*, (Miami, FL, USA), May 2002.
5. T. Fritsch, H. Ritter, and J. H. Schiller, "The Effect of Latency and Network Limitations on MMORPGs: a Field Study of Everquest 2," in *Proceedings of NetGames*, (Hawthorne, NY, USA), Oct. 2005.
6. M. Dick, O. Wellnitz, and L. Wolf, "Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games," in *Proceedings of NetGames*, (Hawthorne, NY, USA), Oct. 2005.
7. A. Wattimena, R. Kooij, J. van Vugt, and O. Ahmed, "Predicting the Perceived Quality of a First Person Shooter: the Quake IV G-model," in *Proceedings of NetGames*, (Singapore), Oct. 2006.
8. C. Chambers, W. chang Feng, W. chi Feng, and D. Saha, "A Geographic Redirection Service for On-line Games," in *Proceedings of the 11th ACM Multimedia Conference*, (Berkeley, CA, USA), Nov. 2003.
9. T. Henderson, "Observations on Game Server Discovery Mechanisms," in *Proceedings of the 4th ACM Network and System Support for Games (NetGames)*, (Braunschweig, Germany), Apr. 2002.
10. G. Armitage, C. Javier, and S. Zander, "Topological Optimisation for Online First Person Shooter Game Server Discovery," in *Australian Telecom Networks and Application Conference (ATNAC)*, (Melbourne, Australia), Dec. 2006.
11. J. Brun, F. Safaei, and P. Boustead, "Server Topology Considerations in Online Games," in *Proceedings of the 4th ACM Network and System Support for Games (NetGames)*, (Singapore), Oct. 2006.
12. K.-W. Lee, B.-J. Ko, and S. Calo, "Adaptive Server Selection for Large Scale Interactive Online Games," in *Proceedings of NOSSDAV*, (Kinsale, County Cork, Ireland), June 2004.
13. B. Wong, A. Slivkins, and E. Sirer, "Meridian: A Lightweight Network Location Service without Virtual Coordinates," in *Proceedings of ACM SIGCOMM Conference*, (Philedelphia, PA, USA), Aug. 2005.
14. B. Wong and E. G. Sirer, "ClosestNode.com: An Open-Access, Scalable, Shared Geocast Service for Distributed Systems," *SIGOPS Operating Systems Review* **40**, Jan. 2006.
15. C. Chambers and W. chang Feng, "Patch Scheduling for On-line Games," in *Proceedings of the 4th ACM Network and System Support for Games (NetGames)*, (Hawthorne, NY, USA), Oct. 2005.
16. C. Chambers, W. chang Feng, S. Sahu, and D. Saha, "Measurement-based Characterization of a Collection of On-line Games," in *Proceedings of the ACM Internet Measurement Conference (IMC)*, (Berkeley, CA, USA), Oct. 2005.
17. S. Zander and G. Armitage, "A Traffic Model for the Xbox Game Halo 2," in *Proceedings of NOSSDAV*, (Stevenson, WA, USA), June 2005.
18. T. Lang, G. Armitage, P. Branch, and H.-Y. Choo, "A Synthetic Traffic Model for Half Life," in *Australian Telecommunications Networks & Applications Conference (ATNAC)*, (Melbourne, Australia), Dec. 2003.
19. T. Lang, P. Branch, and G. Armitage, "A Synthetic Traffic Model for Quake 3," in *ACM SIGCHI Advances in Computer Entertainment (ACE)*, (Singapore), June 2004.
20. J. Faerber, "Network Game Traffic Modelling," in *Proceedings of the ACM Network and System Support for Games (NetGames)*, (Braunschweig, Germany), Apr. 2002.
21. G. Armitage and L. Stewart, "Limitations of using Real-World, Public Servers to Estimate Jitter Tolerance of First Person Shooter Games," in *ACM ACE*, (Singapore), June 2004.
22. M. Claypool and K. Claypool, "Latency and Player Actions in Online Games," *Communications of the ACM* **49**, Nov. 2006.
23. G. Armitage, M. Claypool, and P. Branch, *Networking and Online Games Understanding and Engineering Multiplayer Internet Games*, John Wiley and Sons, Ltd., June 2006. ISBN 0-470-01857-7.