Measurement-based Characterization of a Collection of On-line Games

Chris Chambers *Wu-chang Feng* *Sambit Sahu* *Debanjan Saha*
Portland State University *IBM Research*
{*chambers,wuchang*}@*cs.pdx.edu* {*ssahu,dsaha*}@*us.ibm.com*

Abstract

On-line games are a rapidly growing Internet application. Because of the cost in supporting on-line games and the unpredictable load on servers, companies are moving toward sharing infrastructure for game hosting. To efficiently provision on-line games, it is important to understand game workloads and the behavior of game players. In this paper, we present a comprehensive analysis of a collection of on-line game players and game workloads using data from several sources including: a 13-month trace of an extremely busy game server containing over 2.8 million connections, a two-year trace of the aggregate game populations of over 550 on-line games, and a 4-month trace of a content-distribution network used to deliver games. The key findings from our measurement study are: (1) these gamers are an extremely difficult set of users to satisfy and unless game servers are properly set up and provisioned, gamers quickly choose to go elsewhere, (2) the popularity of these games follows a power law making games difficult to provision at launch time, (3) game workloads are predictable only over short-term intervals, (4) there are significant challenges in hosting games on shared infrastructure due to temporal and geographic synchronization across different games and other interactive applications, and (5) game software updates are a significent burden on game hosting that must be planned for. Our results have implications for both game publishers as well as infrastructure providers.

1 Introduction

On-line gaming is an increasingly popular form of entertainment on the Internet, with the on-line market predicted to be worth over $5 billion dollars in 2008 [1]. As an example of a popular, money-making game, EverQuest [2] has over 450,000 subscribers each paying a monthly fee and purchasing two yearly expansions. Unfortunately for game companies, the success of a game is highly unpredictable. To make matters worse, there are substantial costs in developing and hosting on-line games. As a result, such companies are increasingly exploring shared, on-line hosting platforms such as on-demand computing infrastructure provided by companies such as IBM and HP [3, 4, 5, 6, 7, 8, 9, 10].

In order to judge the feasibility of such an approach, it is important for game and hosting companies to understand how gamers and game workloads behave. Knowing the behavior of players, the predictability of workloads, and the potential for resource sharing between applications allows infrastructure to be tailored to the needs of games. While there has been a substantial amount of work characterizing web and peer-to-peer users and workloads [11, 12], there is very little known about game players and workloads.

In order to provide insight into such issues, this paper examines several large traces of aggregate player populations of a collection of popular games as well as the individual player population of a busy game server. We present a detailed analysis of on-line game players and workloads that targets several key areas which are important to game and hosting providers including:

- *How easy is it to satisfy gamers?*: One of the key issues in providing a successful game is to understand how players connect to servers and how long they play on them. By understanding what players are willing to put up with, game and hosting companies can tailor their infrastructure and content to maximize player satisfaction. For example, one of the challenges with using on-demand computing infrastructure for games is the latency associated with re-purposing a server. It would thus be useful to characterize how patient game players are in connecting to a game before deploying such infrastructure. To this end, we characterize individual player behavior of an extremely popular Counter-Strike game server over a long period of time. Our results show that gamers are an extremely difficult set of users to satisfy and that unless game servers are properly set up and provisioned, gamers quickly choose to go elsewhere.

- *How predictable are game workloads?* Another problem in hosting on-line games is determining the amount of hardware and network bandwidth that is required. Hosting a game is an expensive proposition, costing the game provider more than 30% of the subscription fees in just hardware and bandwidth per month [13]. Hosting is made all the more difficult by variations of popularity as the game moves through its life cycle. Game companies face the provisioning problem both in determining the amount of resources to provide at launch time and in allocating spare resources to support dynamic usage spikes and subscriber growth. Characterizing the diversity and predictability of game workloads allows companies to more accurately provision resources. To this end, we examine the real-time aggregate game player population of more than 550 on-line games, the most popular of which are first-person shooters. Our results show that the popularity of these games follows a distinct power law distribution making the provisioning of resources at launch-time extremely difficult. However, as games mature, their aggregate populations do become predictable, allowing game and hosting companies to more easily allocate resources to meet demand.

- *Can infrastructure be shared amongst game and other interactive applications?* With the advent of commercial on-demand computing infrastructure, it is becoming possible to statistically multiplex server resources across a range of diverse applications, thus reducing the overall hardware costs required to run them. In order for such shared infrastructure to provide any savings, peak usage of applications must not coincide. To characterize the amount of sharing benefit that is available, we examine the usage behavior of a number of popular on-line games and compare them against each other and against the usage behavior of several large distributed web sites. As on-demand infrastructure is distributed, we also examine the client load of a number of servers based on geographic region. Our results show that usage behavior of interactive applications follows strict, geographically-determined, time-of-day patterns with limited opportunities for resource sharing.

Section 2 describes the methodology behind our study. Section 3 analyzes properties of individual gamers. Section 4 describes trends of on-line gaming in aggregate. Section 5 evaluates the potential for multiplexing games and web traffic together, and Section 6 discusses our conclusions.

`cs.mshmro.com` trace	
Start time	Tue Apr 1 2003
End time	Mon May 31 2004
Total connections	2,886,992
Total unique players	493,889

GameSpy trace	
Start time	Fri Nov 1 2002
End time	Fri Dec 31 2004
Total games	550
Total player time	337,765 years

Steam CDN trace	
Start time	Mon Sep 27 2004
End time	Mon Apr 8 2005
Content transferred	6,193 TB
Average transfer rate	3.14 Gbs

Table 1: Data sets

2 Methodology

The study of on-line game usage is typically limited due to the proprietary nature of the industry. To overcome this, we have collected several unique data sets that allow us to analyze properties that have not been possible previously. These data sets include the following:

Individual player data: In order to study the behavior of individual players playing a representative on-line game, we examined the activity of one of the busiest and longest running Counter-Strike servers in the country located at `cs.mshmro.com` [14, 15]. Counter-Strike (a Half-Life modification) is currently the dominant on-line game with the largest service footprint of any game at 35,000 servers and over 4.5 billion player minutes per month [16]. Of all of the active Counter-Strike servers, `cs.mshmro.com` is among the busiest 20 servers as ranked by ServerSpy [17]. The server averages more than 40,000 connections per week, has hosted more than 400,000 unique players within the last year, and has logged more than 60 player years in activity since its launch in August 2001. Table 1 describes the trace collected from the server.

GameSpy aggregate player population data: One problem with measuring on-line game usage is the limited access to game server hosting data. Game companies typically keep the access and usage behavior of their players confidential. There are two factors that enable the measurement of aggregate game player populations, however:

(1) on-line games use a centralized authentication server to keep track of the players that are playing and (2) information on overall player numbers per game is usually exported publicly. Several game portal services collect such player numbers over a large number of games and report the information in real-time. Among these services is the GameSpy network, which provides real-time player population data on individual games in a structured format that can readily collected and analyzed [18]. Currently, there are over 550 on-line games that are being tracked across various genres including first-person shooter games (FPS), massively multi-player on-line role-playing games (MMORPG), real-time strategy games (RTS), card and board games, and sports games. The most popular games tracked by the Gamespy network are from the FPS genre however, and therefore when we refer to gamers we are predominately referring to FPS gamers. To study on-line game population behavior, we have collected a data feed from GameSpy for more than two years since November 2002. Our redundant collection facility periodically samples the GameSpy data every 10 minutes. Note that the availability of the data is sensitive to many factors, including service outages at the portal and our own outages. These outages have been manually removed from the data analysis. Table 1 describes the data set which includes over 50 million measurements and represents more than 300,000 years of player time spent on games over the course of the last two years.

Content-distribution network: One of the common features of on-line games is their ability to dynamically update themselves. To support this feature, many games employ custom, game-specific, content distribution networks that deliver new game content and software patches to clients when needed. One such network is Steam [19], a multi-purpose, content-distribution network run by Valve which is used to distribute run-time security modules as well as client and server software patches for Half-Life and its mods such as Counter-Strike and Day of Defeat. The network consistently delivers several Gbps of content spread across over 100 servers. In order to analyze the resource usage of Steam, we have collected its data feed over the last 6 months, a duration that has seen Steam deliver more than 6 petabytes of data. Table 1 describes the trace collected.

3 Gamers as individuals

It is important for game providers to understand the usage behavior of its players in order to adequately address their needs. In order to study player characteristics, we analyze the trace of `cs.mshmro.com` to track individual gamers throughout their play cycle. Specifically, we track gamers attempting to connect to the server, gamers playing on the server, and the likelihood of a gamer returning to the server.

We first demonstrate that gamers are difficult to please.

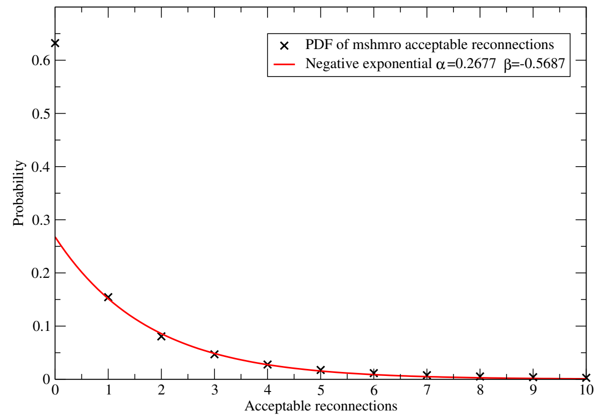


Figure 1: PDF of player impatience based on number of acceptable reconnects

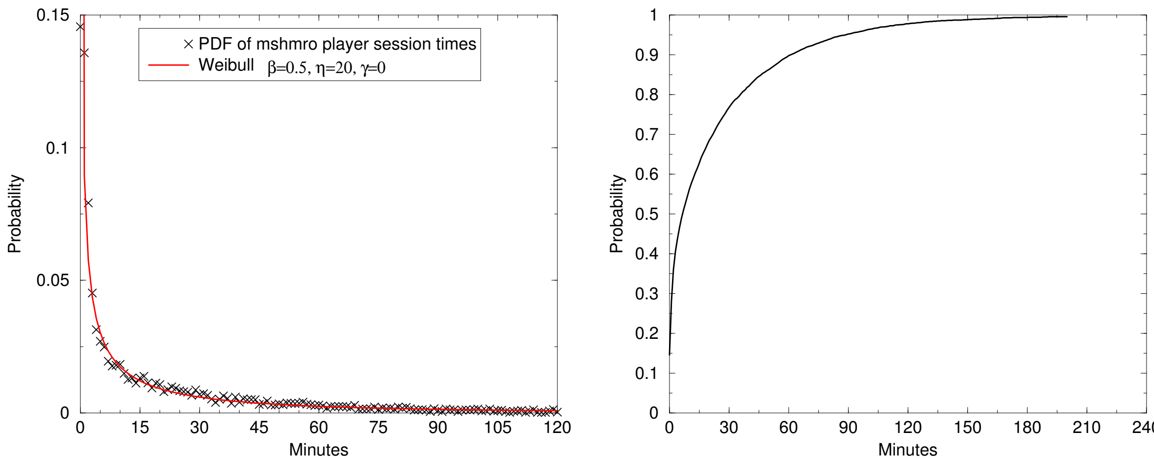
In particular, they 1) have no tolerance for busy servers, often connecting once while the server is busy and never reconnecting again for the entire trace, 2) have very specific gameplay needs and if those needs are not met in the first few minutes of play, their likelihood of continuing to play at the server drops off dramatically, and 3) they often have no loyalty or sense of community tied to a specific server and do not return after playing a handful of times. For those that do return often, we also demonstrate that their session times show a marked decline and their session interarrival times show a marked increase just as they are ready to quit playing on the server altogether.

3.1 Gamers are impatient when connecting

Quantifying the patience of on-line gamers is important for adequate server provisioning. For some Internet applications, such as web-browsing, users are known to be impatient [20]. For others, such as peer-to-peer services such as *Kazaa*, users are very patient [12].

Our trace of `cs.mshmro.com` records successful connections as well as connection attempts, when players connect to the server and are refused service. The latter is extremely common; every day, the server turns away thousands of people. Browsing the trace, it is not unusual to see the same player reconnect to the server several times in a row, waiting for a spot on the server to free up. We operate on the assumption that a player's willingness to reconnect to the same busy server repeatedly is an indication of their patience.

In order to quantify player patience we group each player's connection history into sessions, and consider a session of length N evidence of that player's willingness to reconnect after $N - 1$ connections. Figure 1 shows the probability distribution of acceptable reconnections per



(a) PDF (b) CDF

Figure 2: Session time results for `cs.mshmro.com` trace

player. As the figure shows 73% of the players are unwilling to reconnect to the server enough to play even once. One of the reasons players do not reconnect is that game clients have a "Quick Start" mechanism that many players use. The mechanism works by downloading a list of candidate servers from the master server and cycling through them one by one until a successful session is established. Thus, such clients may not lack patience, but rather are automatically redirected elsewhere. For the rest of the players, however, 13% are willing to reconnect one time on average with the percentage sharply decreasing on successive reconnects. Aside from the first data point, the rest of the graph represents a client's patience in connecting to our busy server and, not surprisingly, can be fit very closely with a negative exponential distribution. As Figure 1 shows, a negative exponential distribution with parameters $\alpha = 0.2677$ and $\beta = -0.5687$ fits the data with a correlation coefficient of 0.998. Players, therefore, exhibit a remarkable degree of impatience with busy game servers.

3.2 Gamers have short attention spans

Using the same trace, we extracted the total session time of each player session contained in the trace. Figure 2 plots the session time distributions of the trace in unit increments of a minute [1]. The figure shows, quite surprisingly, that a significant number of players play only for a short time before disconnecting and that the number of players that play for longer periods of time drops sharply as time increases. Note that in contrast to heavy-tailed distributions reported for most source models for Internet traffic; the session ON times for game players is not heavy-tailed. To further illustrate this, Figure 2(b) shows the cumulative density function for the session times of the trace. As the figure shows, more than 99% of all sessions last less than 2 hours.

Unlike the player patience data, session times can not be fitted with a simple negative exponential distribution. However, the data can be closely matched to a Weibull distribution, a more general distribution that is often used to model lifetime distributions in reliability engineering [22]. Since quitting the game can be viewed as an attention "failure" on the part of the player, the Weibull distribution is well-suited for this application. The generalized Weibull distribution has three parameters β, η, and γ and is shown below.

$$f(T) = \frac{\beta}{\eta}(\frac{T-\gamma}{\eta})^{\beta-1}e^{-(\frac{T-\gamma}{\eta})^{\beta}}$$

In this form, β is a shape parameter or slope of the distribution, η is a scale parameter, and γ is a location parameter. As the location of the distribution is at the origin, γ is set to zero, giving us the two-parameter form for the Weibull PDF.

$$f(T) = \frac{\beta}{\eta}(\frac{T}{\eta})^{\beta-1}e^{-(\frac{T}{\eta})^{\beta}}$$

Using a probability plotting method [22], we estimated the shape (β) and scale (η) parameters of the session time PDF. As Figure 2(a) shows, a Weibull distribution with $\beta = 0.5$, $\eta = 20$, and $\gamma = 0$ closely fits the PDF of measured session times for the trace.

This result is in contrast to previous studies that have fitted a negative exponential distribution to session-times of multiplayer games [23]. Unlike the Weibull distribution which has independent scale and shape parameters, the

[1] Note that a preliminary version of our results here were first reported in a short paper at the NetGames 2003 Workshop [21]

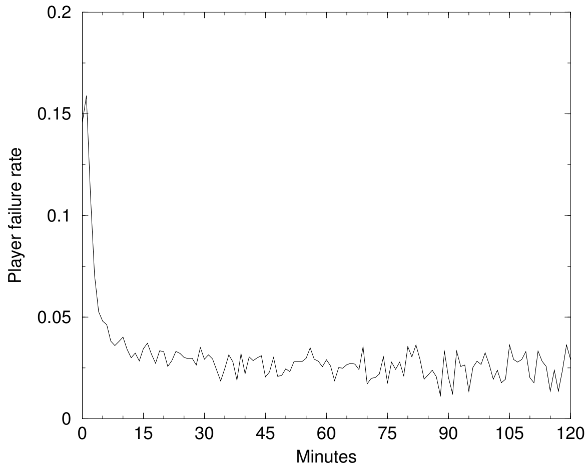


Figure 3: Player failure rates for individual session times for `cs.mshmro.com` trace

shape of the negative exponential distribution is completely determined by λ, the failure rate. Due to the memory-less property of the negative exponential distribution, this rate is assumed to be constant. Figure 3 shows the failure rate for individual session durations over the trace. As the figure shows, the failure rate is *higher* for flows of shorter duration, thus making it difficult to accurately fit it to a negative exponential distribution. While it is difficult to pinpoint the exact reason for this, it could be attributed to the fact that Counter-Strike servers are notoriously heterogeneous. Counter-Strike happens to be one of the most heavily modified on-line games with support for a myriad of add-on features [24, 25]. Short flows could correspond to players browsing the server's features, a characteristic not predominantly found in other games. As with player patience, it may be possible to fit a negative exponential for longer session times. As part of future work, we hope examine this as well as characterize session duration distributions across a larger cross-section of games to see how distributions vary between games and game genres.

3.3 Gamers are not loyal

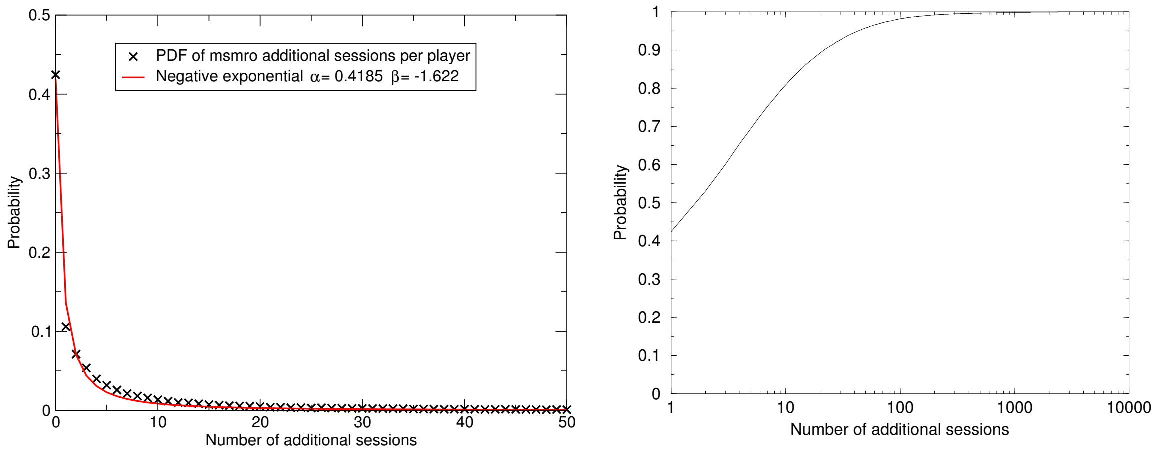
Public-server games such as Half-life provide users with a large choice of servers located all around the world. Gamers can switch between servers as often as they like. Some reasons to continue playing on the same server are simplicity, a known low-latency connection, preference for server options, or a sense of community. It is natural to wonder whether servers continue to serve the same group of clients and to what extent these reasons or others keep clients at a specific server.

Our trace contains the connection records for each client via their unique player identification number (WONID). We quantify loyalty to the server by counting the number of times a player returns to play after a successful playing session. Figure 4(a) shows the probability density function of additional game sessions per player for players who returned at least once to the server while Figure 4(b) shows, on a logarithmic scale, the cumulative distribution. As the figure shows, 42% of the players in our trace returned to play only once and 81% played less than 10 times. On the other hand, the top 1% of loyal gamers return to play hundreds of times (hence the logarithmic scale). It appears that the majority of clients have very little loyalty to public servers, and only a small fraction have grown strongly attached. We hypothesize that, due to a large population of servers to choose from (over 30,000), clients rarely select the same server twice.

3.4 Gamers reveal when they lose interest

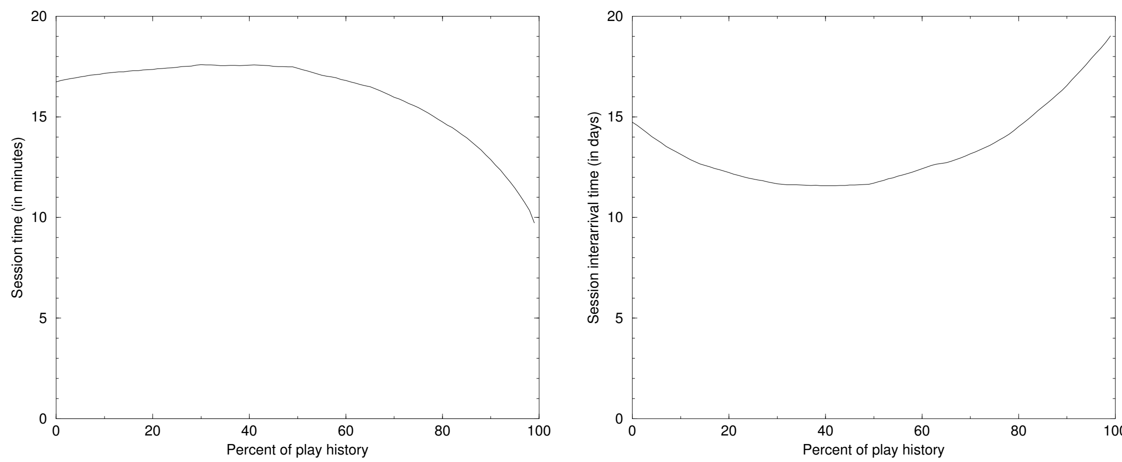
Players of a game have some discretion about how frequently they play a game and for how long. Players often lose interest in a game and cease playing altogether at some point. Before that happens, however, there may be noticeable indications that their interest is waning. Such indications are extremely useful to game providers who can detect waning interest and react to it on a macro level with new content or on a per-player basis via customized incentives for continued play.

We determine the average player interest curve by calculating each player's sequence of play sessions from their first session to their last recorded session. This is a player's *play history*. Since each player may progress through his or her game interest at a different rate, we normalize each of these data sets based on the duration each player is active on the server. We then examine the average session times and session interarrival times of all players throughout their playing careers. Figure 5(a) shows that player session times are relatively constant halfway through their play history and fall off to just more than 50% of the initial session time before the player loses interest completely. Figure 5(b) shows that the time between player sessions is minimized before the halfway point and increases steeply until the player's interest has fully waned. The variance on this data is extremely high, due in part to the fact that players only spend a portion of their time on this single server, and therefore this data is unsuitable for predicting the interest of a given player. However we believe this methodology can be used for games with a centralized session-tracking authority as an early indicator of peaking player interest and that game publishers should use these measurements to trigger the delivery of new content or incentives for the individual player.



(a) PDF (b) CDF

Figure 4: Distribution of sessions per player on the server



(a) Average session time (b) Average session interarrivals

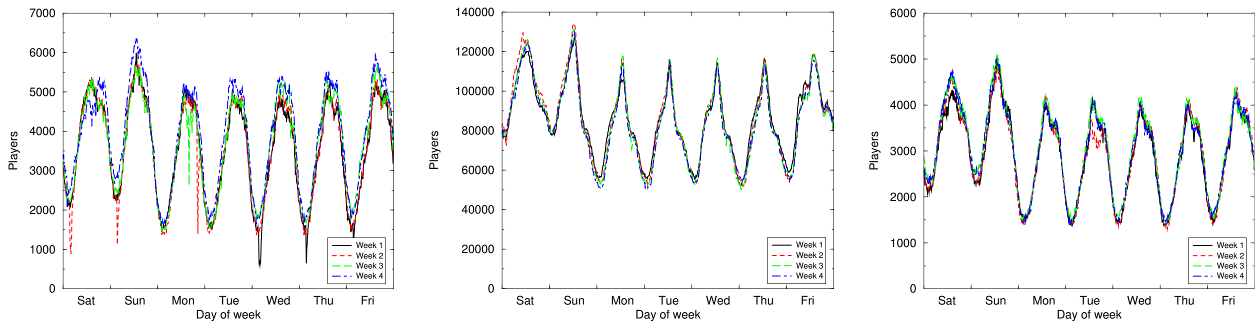
Figure 5: Player behavior throughout their playing careers

4 Game populations

Hosting games is challenging, in part, due to the difficulties of accurate provisioning. Under-provisioning can test gamer patience, while over-provisioning can be costly. We look at two facets of gaming integral to successful game provisioning: overall game popularity and predicting game workloads. We show that (1) there are, and will be, very few extremely popular games, and (2) game workloads are periodic and predictable over short-term intervals.

4.1 Game popularity follows a power-law

To determine the distribution of on-line game popularity, we analyzed a nine-month subset of the GameSpy data set described in Section 2, starting March 1st 2003. Of the games, we consider only the top 50 games, as the remaining games averaged a minimal number of players throughout the trace. To average popularity rankings we first calculated the rank ordering of the games and the number of players at a given rank for each day. Then we averaged these daily rankings over the nine-month period to show the distribution of players across the games regardless of fluctuations in individual game popularity. Figure 6 shows the popularity data on a log-log scale. As the figure shows, this distribution is heavily skewed in favor of the most popular games, with the first ranked game having over ten times the number of players of the next most popular. This distribution of popularity is most similar to a power-law distribution. Power-law distributions are of the form $y = ax^{\lambda}$



(a) America's Army (b) Half-Life (c) Neverwinter Nights

Figure 7: Player load for three popular games over a 4-week period

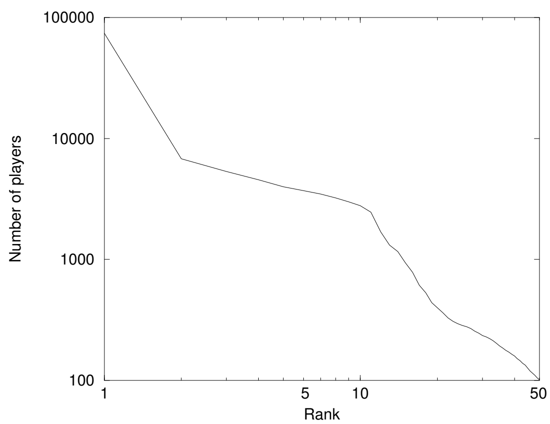


Figure 6: Game popularity distribution averaged over nine months (log scale)

and occur in a number of places including the frequency of words in the English language, the popularity of web pages, and the population of cities. An intuition for these distributions is that whenever choices are made between many options, and each choice affects other choices, the choices tend to pile up on a few popular selections. Games and servers create communities: in selecting one, each player's choice affects and is affected by the popularity and reputation of that game or server. A perfect power-law distribution would graph as a straight line on a logarithmic scale in both the x and y axis. The relatively straight line (correlation coefficient -0.98 for a simple linear regression) demonstrates that the GameSpy data does follow a power law distribution. This distribution has an interesting, albeit unfortunate, implication for provisioning server resources for on-line games: the host must plan for several orders of magnitude of change in popularity (and therefore resources) in either direction. As a result, this indicates that on-demand infrastructure can significantly reduce the costs and risks of launching and hosting on-line games.

4.2 Game workloads have varying degrees of predictability

Accurately predicting game workloads allows game hosting providers to allocate the appropriate amount of resources for a game. In order to determine whether this is feasible, we analyze the GameSpy trace for different sets of games. Specifically, we investigate whether any simple trends or patterns can be used to accurately predict the game workload, whether the workload is stable and if so, over what time scale.

4.2.1 Game workloads exhibit predictable daily and weekly changes

Intuitively, it is reasonable to assume that usage is strongly tied to the daily and weekly activities of players. Figure 7 shows the global player population of four consecutive weeks starting from 3/1/2003 for three popular games: America's Army, Half-Life, and Neverwinter Nights. As expected, the figure shows that the workload has regular daily cycles and that over this one month period the workload does not vary significantly from week-to-week. In fact, for all three games, the trends as well as the maximum and minimum points match up at identical points in time during the week. We observe similar results over other parts of the year with the only anomalies caused by service outages and by holidays. To further demonstrate the cyclical nature of gaming workloads, we take one year's worth of game server load samples across a variety of games and plot the Fast Fourier Transform (FFT) of the data. The FFTs have been scaled so that they can be plotted together. As Figure 8 shows, the FFT contains strong peaks at the 24-hour cycle for each of the games. There is also a significant peak at the 168-hour (one week) cycle for two of the games as well. This corresponds to an increase in player usage on

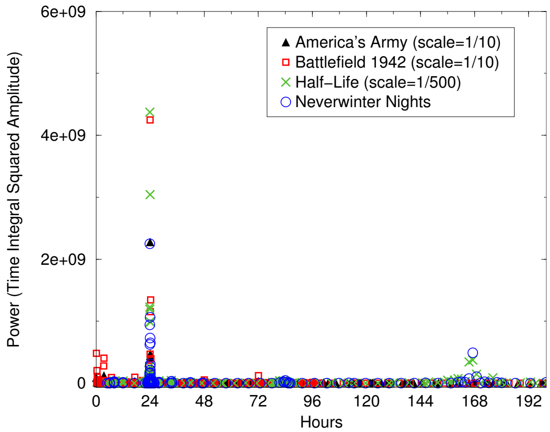


Figure 8: FFT of the player load from four games over one year.

the weekends during some parts of the year. Papagiannaki et. al use wavelet multiresolution analysis (MRA) on another long-term data series [26], and model their series as a 12-hour and 24-hour cycle plus a trend. We were unable to apply this technique however, due to the reliance of wavelet MRA on resolutions that are factors of two apart. The difference between our two cycles is seven.

In order to quantify the week-to-week variation of game workloads, Figure 9 shows distribution of week-to-week load changes of the top 5 games during 2004: Half-Life, Battlefield 1942, Medal of Honor: Allied Assault, America's Army, and Neverwinter Nights. Figure 9(a) plots the distribution of instantaneous load changes between identical points in time of consecutive weeks, while Figure 9(b) plots the change in average daily load between the same day of the week of consecutive weeks. Finally, Figure 9(c) plots changes in maximum daily load between the same day of the week of consecutive weeks. The figures fit a 't' location-scale distribution, which has three parameters, a scale parameter $\sigma > 0$, a location parameter μ, and a shape parameter $\nu > 0$. The density function for this distribution is as follows:

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sigma\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}(\frac{\nu+(\frac{x-\mu}{\sigma})^2}{\nu})^{-\frac{v+1}{2}}$$

Note that if x is 't' location-scale distributed, $\frac{x-\mu}{\sigma}$ is Student's 't' distributed with ν degrees of freedom. As illustrated in Figure 9, we find a very good fit for all the three plots. Based on this observation, we draw two main conclusions with regard to resource usage:

- As the figures show, almost all week-to-week load variations are under 10% of the previous week's workload. Such behavior makes it relatively easy for game and infrastructure providers to provision and predict resource usage on a weekly basis.
- The above distribution fitting of load variations indicates that it is feasible to model the week-to-week load variations using such standard distributions. We are exploring the feasibility of online parameter estimations for using this model in resource provisioning.

4.2.2 Game workloads exhibit unpredictable long-term fluctuations

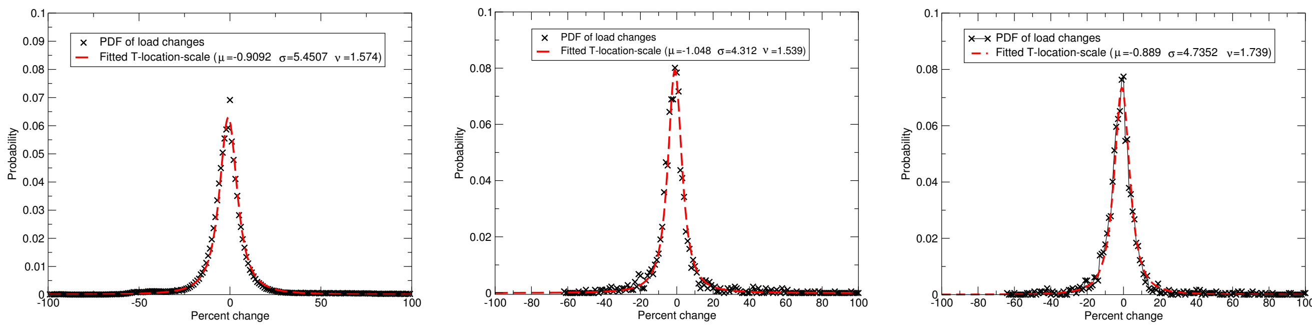
While the daily and weekly cycles in server load are clear, the duration of our trace allows us to examine longer term cycles. We examine the trend of three games of similar popularity as well as the trend of the most popular game, Half-life, over the period of just over two years. We compute the trend as the moving average of the data with a window size of one week. Figure 10 shows the trends of the respective games. The underlying trend of these games does not reveal periodicities on a monthly timescale, and the limits of our trace prevent us from drawing any strong conclusions about annual cycles. There are several points in trace where the games appear to be synchronized, but the explanation for the concurrent peaks or valleys is not necessarily predictable. We observe peaks in all games near the Christmas season, but, for example, all four games experience a drop during the unpredictable weeks of the *Sobig* virus [27].

5 Potential for multiplexing gain

With the movement toward hosted game services [28, 29] as well as on-demand computing infrastructure for games such as Butterfly.net [30], there has been a great deal of interest in reducing the cost of running game servers by sharing server resources dynamically across multiple games and applications. We explore two likely scenarios: hosting multiple games on the same servers, and hosting web sites along with game servers. In addition, we study the usage behavior of a content-distribution network for supporting games. Our results show that there are significant challenges in multiplexing interactive applications on the same server infrastructure and that only limited opportunities for reducing peak resource usage exist.

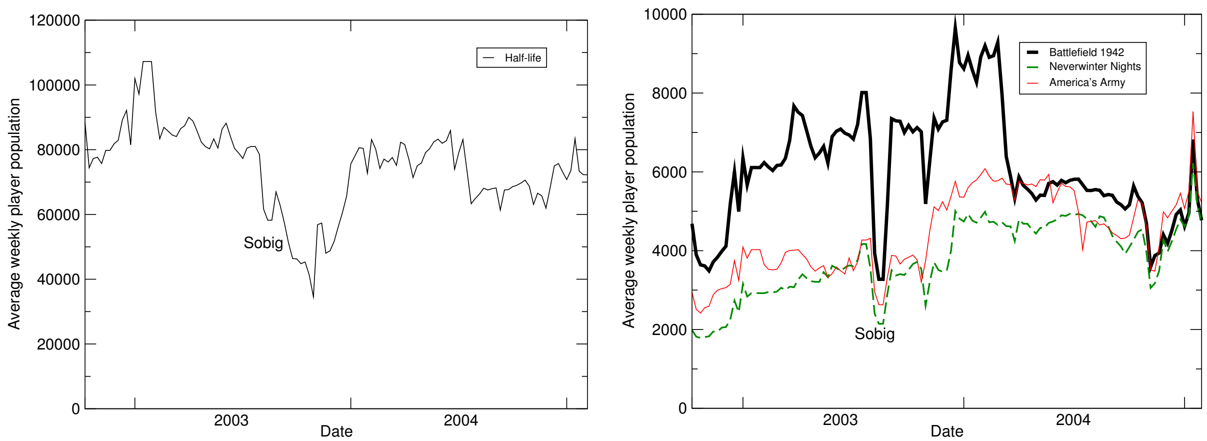
5.1 Game workloads are synchronized

There are two ways games can be multiplexed with each other. One way would be to coarsely and statically assign physical servers to particular games based on the popularity of the game. Results from Section 4 clearly show that this can provide a lot of benefit for game companies. Another



(a) Instantaneous (b) Mean (c) Maximum

Figure 9: Week-to-week PDF of percent load changes for the top 5 games of 2004



(a) Half-Life (b) Other games

Figure 10: Population trends for Half-life and other games with an averaging window of one week

Game	Average number of players
Half-Life	80324
America's Army	5791
Battlefield 1942	5402
Neverwinter Nights	4579

Table 2: Mean player populations for week of May 23, 2004

way would be to dynamically re-allocate servers based on instantaneous demand for a particular game. An implicit assumption that gives value to the latter method is that different games have usage patterns that are substantially different. Thus, rather than have each game provision server resources based on the peak usage of their game, server resources would be provisioned for the global peak.

In order to investigate the extent to which different games can be multiplexed with each other, we examined the aggregate player populations of four popular games. The games examined include FPS games (Half-Life, Battlefield 1942, and America's Army), as well as an MMORPG (Neverwinter Nights). Player populations of these games were collected over a one week period (Sunday May 23, 2004 to Saturday May 29, 2004) from the GameSpy trace. In order to compare the games directly, independent of their popularity, each game's population data was normalized by the mean population for that particular game during the week. Table 2 lists the mean player populations for the four games examined. Figure 11 plots the normalized player loads for the four games during the one week period. As the figure shows, player populations fluctuate significantly based on the time of day from lows close to half of the mean to peaks close to twice the mean. In addition, populations across games have peaks in close proximity to each other, making it difficult to achieve significant statistical multiplexing gain between different games. Finally, as indicated in the FFTs from Figure 8, games show

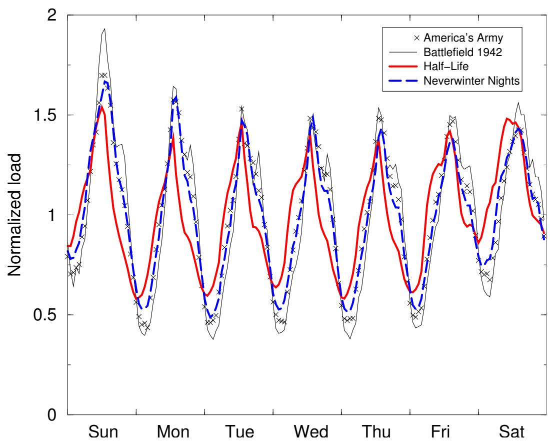


Figure 11: Aggregate normalized load across four popular games for week of May 23, 2004

slight peaks on the weekends with slightly more players on-line than during the week.

5.2 Games and interactive application workloads are synchronized

While Section 5.1 shows the difficulty in obtaining statistical multiplexing gain between different games, on-demand computing infrastructure could still be useful for multiplexing between other applications such as web servers. In order to examine this, we obtained web server logs over a week for three commercial sites. The sites included those for a North American cereal manufacturer, a North American credit card company, and an international beverage manufacturer. Table 3 describes the traces of the web servers, all from the week of August 13, 2001. The servers themselves were located in geographically distributed data centers and the individual logs from each site were aggregated and sorted into a single log file. Using these traces, we plotted the normalized load for the web server against the normalized global aggregate load of Half-Life during the same week in August 2004.

As Figure 12 shows, workloads for web and on-line games share similar daily periodic peaks. This particular week of game traffic does not have a strong weekend rise (perhaps due to being from the summer), but the web traffic does slump during the weekends as Figures 12(a) and 12(b) show. Interestingly, Half-life shows considerably less variance than the North American websites, but similar variance to the international beverage manufacturer website. Intuitively, it makes sense that applications and web sites with global usage patterns are more consistently busy and have less daily variance. Due to the international popularity of Half-Life, its usage pattern is quite similar to that of the

North American cereal manufacturer	
Start time	Mon Aug 13 2001
End time	Sun Aug 19 2001
Total requests	10,368,896
Content transferred	59.6 GB

North American credit card company	
Start time	Tue Aug 14 2001
End time	Mon Aug 20 2001
Total requests	112,590,195
Content transferred	366.4 GB

International beverage manufacturer	
Start time	Tue Aug 14 2001
End time	Sat Aug 18 2001
Total requests	11,932,946
Geographically resolvable	11,829,429
Content transferred	51.1 GB

Table 3: Web site logs for week of August 13, 2001

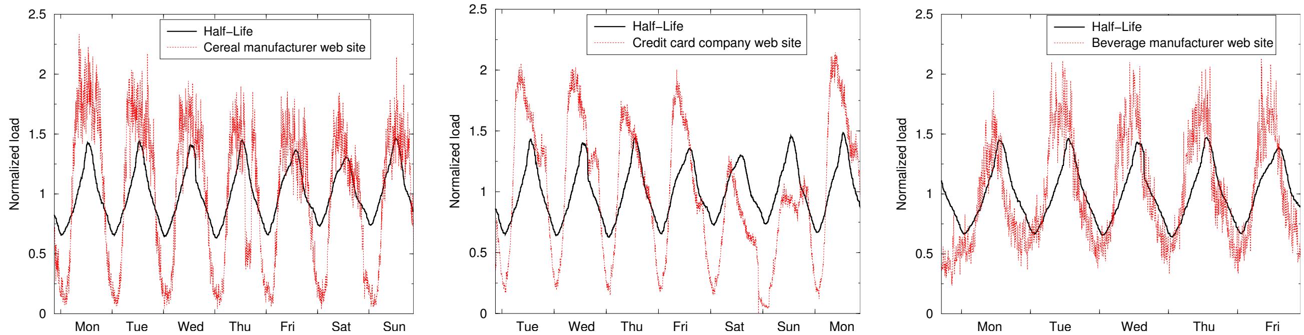
Total connections	71,253
Geographically resolvable	30,226
From North America	9,414
From Asia	9,814
From Europe	8,788
From other continents	2,210

Table 4: Connection data for `cs.mshmro.com` for week of May 23, 2004

international beverage company's web site. Overall, these results indicate that infrastructure sharing between applications during the week will have a somewhat limited benefit with some potential for multiplexing gain during the weekends and during the "off hours" for geocentric applications.

5.3 Games exhibit strong, diurnal geographic patterns

One of the salient features of globally distributed, on-demand computing infrastructure is that it can easily shift resources geographically close to where the demand is coming from. Intuitively, it makes sense that a predictable, diurnal pattern drives global resource consumption and hence, the provisioning of server resources. This is especially the case for applications that require human participants such as games. To study this phenomenon, we examined a one-week period of `cs.mshmro.com` (Sunday May 23, 2004 to Saturday May 29, 2004). Using this log and a commercial geographic IP address mapping tool [31], the location of each player connecting was resolved. As Ta-



(a) North American cereal manufacturer

(b) North American credit card company

(c) International beverage manufacturer

Figure 12: Aggregate normalized load between Half-Life and commercial web sites

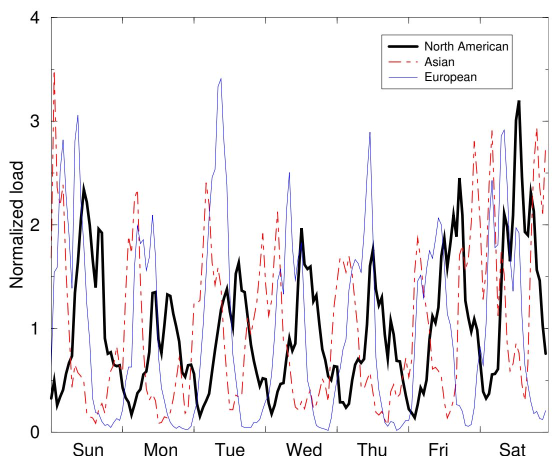


Figure 13: Aggregate normalized load per-continent for `cs.mshmro.com`

ble 4 shows, a significant portion of the load is from outside of North America. Using the resolved connections, the per-continent load normalized by the mean connection arrival rate was plotted. As Figure 13 shows, each continent shows a predictable, diurnal pattern of activity with the only difference being a time-zone shift. It is interesting to note that in contrast to the Half-Life aggregate load and international beverage company web site load (Figure 12(c)), the per-continent load of `cs.mshmro.com` exhibits a large variance similar to the North American web site loads shown in Figures 12(a) and 12(b). We hypothesize that when the usage patterns of international services are broken out into individual regions, the resulting load variances are similar to those of regional servers such as the cereal manufacturer and the credit card company.

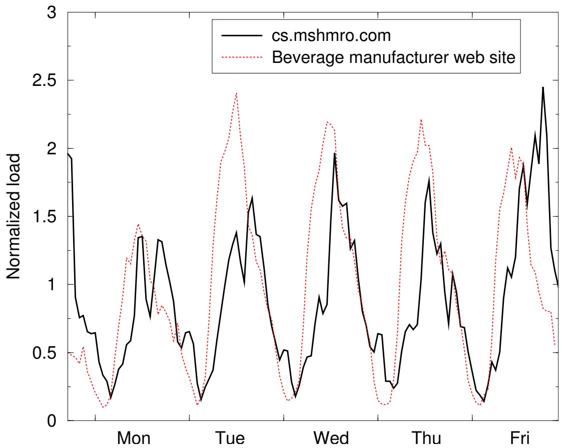
To test this hypothesis, we compared the per-continent load between `cs.mshmro.com` and the international beverage company web server trace [2]. Figure 14 shows the per-continent, normalized load of the game and web server for North America and Europe. The loads from other continents show similar results. As expected, the per-continent load fluctuations and variance are similar to those found in the two regional web sites. The figure also shows that usage of both applications are highly synchronized when broken down into geographic regions. The degree of synchronization thus limits the benefits that geographically distributed, on-demand computing infrastructure has on interactive applications such as games and web.

5.4 Game updates significantly impact resource usage

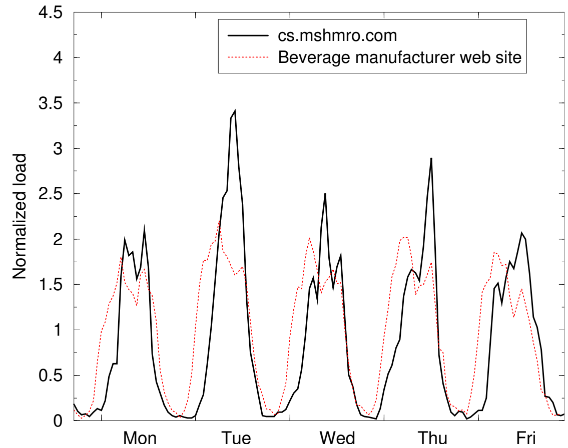
The infrastructure required to host on-line games must also account for the mutability of the games over time. Software patches to fix bugs, prevent cheats, and deliver new content to end-users are an expected component of many on-line games. These patches can vary greatly in size, from a few bytes to several gigabytes. Understanding the impact of these patches on hosting, and adequately provisioning for them is an important part of supporting on-line games. We use the trace of the Steam content delivery network to examine this aspect of games. Our Steam trace includes the initial download of the popular FPS game *Half-Life 2* as well as a number of sizable content updates for both clients and servers.

The Steam network is utilized for both player authentication and content distribution. Players are authenticated to Steam for each game session, via the download of an authentication module. Content is distributed to players (and

[2] Note that in comparing the geographical resolution data of Tables 3 and 4, a much larger percentage of the IP addresses in the beverage company trace is resolvable. This is due to the fact that the trace (and the set of IP addresses in it) is much older, giving services such as GeoBytes more time to identify their locations



(a) North America



(b) Europe

Figure 14: Normalized load for `cs.mshmro.com` and the international beverage company website

servers) via Steam at irregular intervals and irregular sizes. These two functions are not distinguished in the data set we have collected. However, we can differentiate them by utilizing the GameSpy dataset, which tracks player load, by assuming that player load and game authentication are linearly correlated.

As a way of validating that the Steam data and the GameSpy data are tracking the same thing (i.e. player load), we consider a week without a Steam update. Figure 15 shows a scatter plot of Steam data (in megabits per second) versus GameSpy data (in players), and the least-squares fit line. The correlation coefficient for this week is 0.86, indicating a roughly linear relationship. We attribute the inexact nature of the correspondence to small changes in the size of the authentication module and sampling error.

We use the GameSpy dataset to subtract the authentication data from Steam and focus on the bandwidth requirements of a patch. Figure 16 shows a two week period of Steam activity, with a single patch occurring three days into the period. Also graphed is the authentication data component, computed from the GameSpy dataset with a ratio of players to megabits/second of 1 to 0.0291. By integrating these two signals and subtracting, we estimate the patch burden on Steam for this patch to be 129.7 terabytes, which is 30% of that week's total load including authentication.

We use this same methodology on four patches delivered during our trace, and chart the bandwidth impact of the patches over a two-week period in Figure 17. Three anomalies deserve explanation: patch *p3* is cut short of the full two week period analysis because of the release of *p5*, patch *p2* shows a rise in bandwidth after one week due to erroneous player data from GameSpy, and (according to Steam's press releases) the two weeks of patch *p7* contain numerous patches. One question to address is how long it

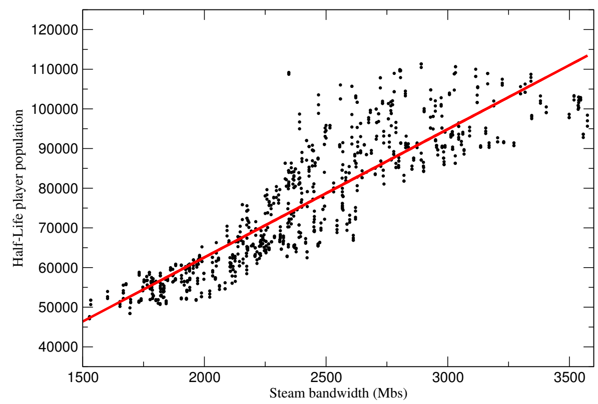


Figure 15: Half-Life player population versus Steam CDN usage

takes to deliver a patch: the cumulative distribution function (CDF) of the patch delivery data in Figure 18 shows that 80% of the load occurs in the first 72 hours for the three single-patch traces, whereas the various patches in trace *p7* are delivered throughout a two-week period.

Our observations on patch distribution bring up several issues. We believe content delivery for games is a significant burden that must be provisioned for, as it can greatly increase the hosting bandwidth requirement. At this point, however, it is unclear what the optimal strategy would be for delivery and scheduling. Our inital observations are that to avoid the stacking effect seen in Figure 18, content should be spaced for delivery such that the bulk of each patch is delivered before the next patch begins. Further, if minimizing the combined content and authentication load is a goal, then patches should be released at the lowest peak

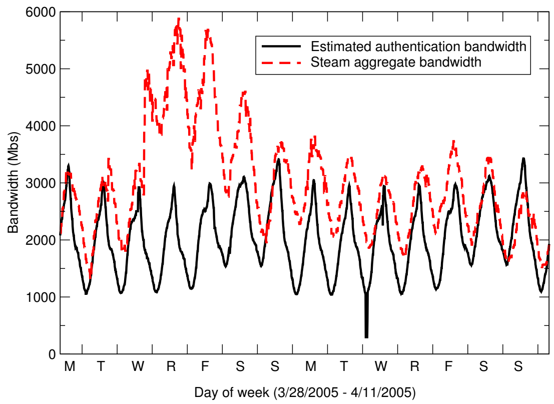


Figure 16: Steam bandwidth during a patch release

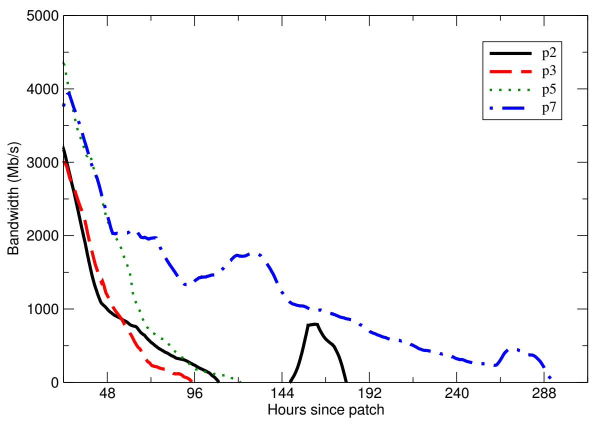


Figure 17: Excess bandwidth consumed by users downloading patches via Steam

in the weekly and daily cycle. For example, a patch released Monday evening may potentially miss the daily afternoon peak as well as the weekend peak. As part of future work, we plan on examining the proper scheduling of patches based on measured game workloads.

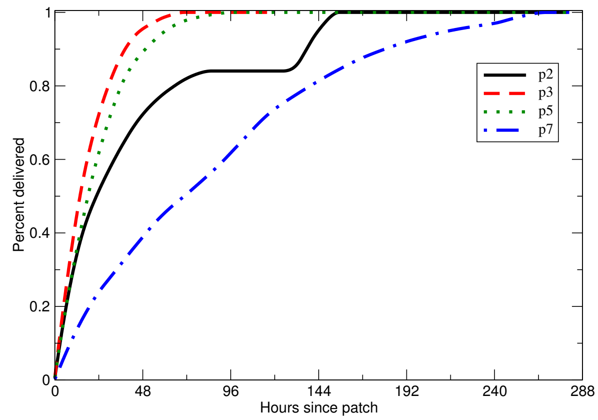


Figure 18: Cumulative distribution function of patch data.

6 Conclusions

On-line gaming is an increasingly popular form of entertainment on the Internet. Unfortunately, effectively hosting on-line games is a difficult, expensive proposition made more onerous by the lack of workload models for games or known characteristics of gamers. Due to the unpredictable nature of the popularity of a game, combined with the high barrier to entry for hosting, a number of academic and industry projects have focused on providing a shared on-demand infrastructure to solve the hosting problem.

To understand the benefits of such infrastructure, this paper presents a comprehensive analysis of a collection of on-line game players and game usage data from a number of unique sources, mostly biased towards the FPS genre. Our results show that gamers are difficult to satisfy throughout the gameplay process: they are likely to leave and never return if they can't connect, they are likely to leave within the first few minutes if they don't enjoy the server's characteristics, and they are unlikely to become loyal to a server. In addition, the popularity of this collection of games follows a power-law distribution, with a small number of games having orders of magnitude more players than the rest. This makes resource provisioning very difficult for the initial release of a game when popularity has not been established and provides a promising area where shared hosting can provide benefit. Although initial provisioning is difficult, our results also show that once established, game workloads are relatively stable from week to week, allowing game providers to more easily allocate resources to meet demand. In addition, we determine that game workloads are synchronized amongst themselves and other interactive applications and that they follow strong diurnal, geographic patterns. Such synchronization makes it difficult to obtain statistical multiplexing gain between games and other interactive applications when using shared infrastructure. Finally, we show that game software updates provide a significant burden on game hosting and must be scheduled and planned for accordingly.

References

[1] DFC Intelligence, "Online Game Market is Growing but Making Money is Difficult," `http://www.dfcint.com/news/prjune252003.html`.

[2] Sony Online Entertainment, "Sony Online Entertainment Ships Everquest: Gates of Discord,"

http://sonyonline.com/corp/press_releases/020904_GOD_ships.html, 2004.

[3] IBM Corp., "On demand business," http://www.ibm.com/ondemand.

[4] IDC, "HP utility data center: Enabling enhanced data center agility," http://www.hp.com/large/globalsolutions/ae/pdfs/udc_enabling.pdf, May 2003.

[5] Sun Microsystems, "N1 Grid – introducing just in time computing," http://wwws.sun.com/software/solutions/n1/wp-n1.pdf, 2003.

[6] GGF, "Global Grid Forum," http://www.ggf.org.

[7] IBM Corp., "Tivoli intelligent thinkdynamic orchestrator," http://www.ibm.com/software/tivoli/products/intell-orch, 2004.

[8] E. Manoel et al., *Provisioning On Demand: Introducing IBM Tivoli Intelligent ThinkDynamic Orchestrator*, IBM International Technical Support Organization, December 2003, http://www.redbooks.ibm.com.

[9] D. Saha, S. Sahu, and A. Shaikh, "A Service Platform for On-Line Games," in *NetGames*, Redwood City, CA 2003.

[10] A. Shaikh, S. Sahu, M. Rosu, M. Shea, and D. Saha, "Implementation of a Service Platform for Online Games," in *NetGames*, August 2004.

[11] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web Traffic: Evidence and Possible Causes," in *Proceedings of ACM SIGMETRICS*, May 1996.

[12] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer Workload," in *Proceedings of ACM SOSP*, October 2003.

[13] Online Game Publisher, "Private Communication," 2004.

[14] mshmro.com, "Counter-strike Server," http://www.mshmro.com/.

[15] W. Feng, F. Chang, W. Feng, and J. Walpole, "Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server," in *Proc. of the Internet Measurement Workshop*, November 2002.

[16] AMDZone, "Valve Releases Hammer Port of Counter-Strike Server," http://www.amdzone.com/releaseview.cfm?ReleaseID=1050, 2003.

[17] ServerSpy, "ServerSpy.Net: World Server Ranks," http://www.serverspy.net/site/serverranks/, 2004.

[18] GameSpy Industries, "GameSpy: Gaming's Home Page," http://www.gamespy.com/, 2002.

[19] Valve, Inc., "Steam," http://www.steampowered.com/, 2005.

[20] L. Catledge and J. Pitkow, "Characterizing Browsing Strategies in the World-Wide Web," *Computer Networks and ISDN Systems*, vol. 27, no. 6, pp. 1065–1073, 1995.

[21] F. Chang and W. Feng, "Modeling Player Session Times of On-line Games," in *NetGames 2003*, May 2003.

[22] ReliaSoft Corporation, "Life Data Analysis and Reliability Engineering Theory and Principles Reference from ReliaSoft," http://www.weibull.com/lifedatawebcontents.htm, 2003.

[23] T. Henderson and S. Bhatti, "Modelling User Behavior in Networked Games," in *ACM Multimedia*, 2001, pp. 212–220.

[24] Half-Life Admin Mod Developers, "Half-Life Admin Mod Home," http://www.adminmod.org/.

[25] AMX Mod Developers, "AMX Mod Server Plugin," http://amxmod.net/.

[26] D. Papagiannaki, N. Taft, Z. Zhang, and C. Diot, "Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models," in *Proc. IEEE INFOCOM*, 2003.

[27] CNN, "SoBig.F Breaks Virus Speed Records," http://www.cnn.com/2003/TECH/internet/08/21/sobig.virus, 2003.

[28] Microsoft Corporation, "Xbox Live," http://www.xbox.com/live, 2003.

[29] Electronic Arts, Inc., "EA.com," http://www.ea.com/, 2003.

[30] Butterfly.net, Inc., "Butterfly Grid Solution for Online Games," http://www.butterfly.net/, 2003.

[31] Geobytes, Inc., "Geobytes Home Page," http://www.geobytes.com/, 2003.