Modelling user behaviour in networked games

Tristan Henderson and Saleem Bhatti
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
UK
{T.Henderson,S.Bhatti}@cs.ucl.ac.uk

June 25, 2001

Abstract

In this paper we attempt to gain an understanding of the behaviour of users in a multipoint, interactive communication scenario. In particular, we wish to understand the dynamics of user participation at a session level. We present wide-area session-level traces of the popular multiplayer networked games Quake and Half-Life. These traces were gathered by regularly polling 2256 game servers located all over the Internet, and querying the number of players present at each server and how long they had been playing. We analyse three specific features of the data: the number of players in a game, the interarrival times between players and the length of a player's session. We find significant time-of-day and network externality effects in the number of players. Player duration times fit an exponential distribution, while interarrival times fit a heavy-tailed distribution. The implications of our findings are discussed in the context of provisioning and charging for network quality of service for multipoint and multicast transmission. This work is ongoing.

1 Introduction

In spite of being an active research area for over a decade, multicast has yet to see large-scale deployment. This may be due to a number of factors, such as the lack of compelling multicast applications, or the lack of a method for multicast service providers to charge for a multicast service [10]. We are in the process of developing a pricing scheme which allows efficient and predictable charging for multicast (initial details of this scheme are described in [15]). One problem with attempting to charge for multiple-source applications, however, is the need for predictable prices. If users share the cost of a multiuser transmission, and the number of users changes as users join and leave the session, the price paid per user will also change. We therefore need to understand how users behave in multiuser scenarios if we are to engineer pricing schemes that will provide stable and predictable prices. Models for user behaviour are also useful in designing pricing schemes for maximising objectives such as aggregate utility or network utilisation, and for understanding and providing for network Quality of Service (QoS).

In a somewhat "chicken and egg" situation, the limited deployment of multicast also means that there are few sources of data from which a model for user behaviour can be determined. A feature of our intended pricing scheme is that it should be independent of the underlying network protocols. Thus, there is no reason why a model for multipoint user behaviour need be determined from IP multicast sessions. From an end-user viewpoint, there is no functional difference between an IP multicast session and several unicast streams, and so user behaviour should be similar in both of these multipoint situations. There may be a difference in cost and this may be used as an incentive for the use of multicast.

With the removal of the absolute requirement for native IP multicast sessions in order to provide multipoint communications, the problem of creating a model for multipoint behaviour becomes more tractable. There are many existing, and popular, multipoint applications that use unicast routing, for example, online chat applications such as Internet Relay Chat (IRC), or multiuser networked games such as *Quake*. We have chosen to examine the latter to determine user behaviour, and thus to determine the requirements for pricing this behaviour and provisioning network resources.

This paper is structured as follows. We discuss our motivation for choosing games as an application and note previous work in Section 2. Section 3 describes the methodology used for gathering data and summarises our results. Sections 4, 5 and 6 analyse three particular aspects of the data, namely session membership, session duration and user interarrival times respectively. Finally, Section 7 concludes the paper and discusses possibilities for future research.

2 Motivation

Multiplayer networked games are contributing to an increasingly large proportion of network traffic [22]. Such network usage is likely to increase further now that consoles such as the Sega Dreamcast and Sony Playstation 2 feature Ethernet and modem connections. Games players are already willing to pay extra to get an improvement in their playing experience, as evidenced by specialist gaming hardware such as joysticks, mice, mousepads and even furniture. Game publishers have proposed charging players per-game via network delivery, rather than the current practice of charging a one-off fee for the software [25], or by charging a fee per game with the opportunity for players to win money or prizes [29]. More interesting, from a networking point of view, is the existence of modems marketed as being specially optimised for games [1], and software designed to determine network characteristics of potential games servers such as delay [13]. These developments indicate that games players are interested in network QoS, and would be willing to pay for the ability to improve it.

The games that we study here are of the type commonly referred to as FPS (First Person Shooter) games. Players connect to a central server using unicast UDP (or occasionally TCP) flows. The maximum number of players that can connect to a server is set arbitrarily by the server administrator, according to the amount of network traffic and CPU time they wish the game server to consume (for the games studied here, this figure is typically set to 16 or 32 players). Players' actions are transmitted first to the central server, which calculates and maintains the overall state of the game and then transmits this state back to the players. The general objective of most of these games is to explore a common virtual world and kill as many of the other players as possible.

2.1 Previous work

Multicast sessions on the MBone are studied by Almeroth and Ammar [2]; these sessions are all single-source and perhaps do not reflect the different dynamics of multiple-source applications. There is a long history of network and Internet traffic analysis (see [24] for a survey). The majority of this, however, looks at packet-level and network-level traces. In particular, Bangun *et al.* [3] and Borella [5] both study the traffic patterns of multiplayer games, but do not examine session-level user dynamics, and limit the studies to local area network traffic only. Although there may be interesting relationships between the data at the packet and session levels, for instance in terms of self-similarity, we do not consider these in this study, but leave them for possible future work.

3 Methodology

Almeroth and Ammar [2] show that the monitoring of IP multicast sessions is possible through joining a session and then watching other session members join and leave. This is impractical for networked games, however, since to join a game implies participation. As most people are only capable of playing one game at a time, and only for a certain number of hours a day, this limits the scope of any data collection. Although it is possible to simulate a user through a script or program, such "bots" are frowned upon by many game server operators and generally lead to the user in question being barred from that server. Furthermore, there is a data integrity problem in that user behaviour might depend on the number of players in a game, and so by joining a game to monitor it, we might affect the results.

Some game servers offer a querying mechanism, whereby specific variables about game status can be retrieved. Since joining and continuously monitoring games seemed impractical, polling and querying games servers at regular intervals was determined to be the next best option. By polling servers and determining the number of players at each poll, an approximation of user behaviour can be obtained. Many networked games also allow the querying of such variables as players' nicknames and the amount of time that they have been playing, and so the duration of each users' session can also be estimated. The accuracy of this method depends on the frequency of polls. If the polls are too far apart, then any users who join and leave between polls will be missed. If the polls are too frequent, the amount of network traffic might have an effect on the servers and perhaps affect user behaviour.

Data were collected using the QStat tool [27], which is a program designed to display the status of games servers, and which supports a large number of online multiplayer games. Of these games, the game *Half-Life* [14] was determined to be the most popular game, and was also one of the games which supports the reporting of a player's connection time, and so it was chosen to concentrate on players of this game.

A list of 2193 IP address/port pairs[1] of machines running the *Half-Life* daemon was obtained from a "master server" at `half-life.west.won.net`. This list is composed from submissions by server administrators and/or automatic registration by servers (depending on the game). This list may also be queried by users through the application itself, or through the use of some of the aforementioned programs for determining the closest or quickest-responding game server.

Servers were polled using QStat at regular intervals (Figure 1). At each poll, the number of players, their chosen nicknames and the number of seconds that each player had been connected were retrieved (example output from QStat is shown in Figure 2). If a response was not received from a server, group membership was assumed to be the same as at the previous successful poll. Since polling took place at the application level, we could not detect such events as unsuccessful join attempts, as these do not register in the game. We were also limited in that polling takes place from a central machine at UCL, and so any network failures that existed solely between UCL and the game servers (but not between the game server and the players) would affect our results.

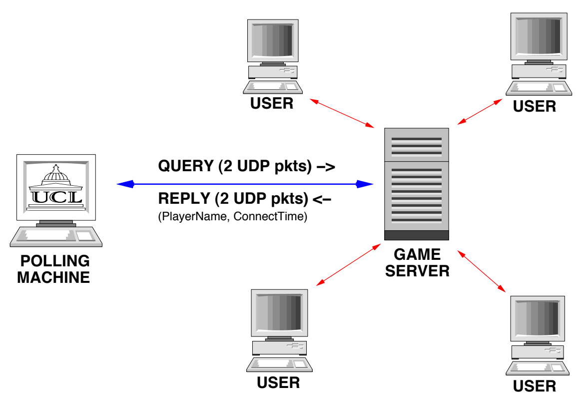


Figure 1: Data gathering setup

	Servers	*Game*	*Frequency*	*Duration*
O-I	2193	Half-Life	30min	1 week
O-II	35	Half-Life	5min	3 days
O-III	22	Quake	5min	1 week
O-IV	3	Half-Life	5min	2 months
O-V	3	Quake III Arena	5min	2 months

Table 1: Observations

[1] It is not uncommon for a single machine to run several servers on different ports; of our list of 2193 servers, there were 1725 unique IP addresses.

```
NAME: Merlin TIME:  5710
NAME: [F.u.T]The_LAW TIME:  5728
NAME: MagNETo [FH] TIME:  2176
NAME: TomiN TIME:  2409
NAME: [DEM] Guybrush T. TIME:  8575
NAME: [.HoF.]Ben Kenobi TIME:  1177
NAME: [Thug]Tosh TIME:  142
NAME: TDMT_Silvan TIME:  1540
NAME: Gulzak TIME:  874
NAME: [DBK]HannibalTC TIME:  954
NAME: [STANDARD] Kill Demon TIME:  1085
NAME: -=Phoenix=- TIME:  5593
```

Figure 2: Example QStat output

Several sets of observations were taken; the differences between these, and the labels that are used in this paper to refer to them, are shown in Table 1. The first set O-I used the master list of *Half-Life* servers. From this, the 35 most popular servers were selected for more detailed observation over one weekend in O-II.

Set O-III used 22 *Quake* servers, the addresses of which were also obtained by querying a master server. *Quake* is an older game, introduced in 1996, which is why the number of servers is so much lower than *Half-Life*, which first went on sale in 1998. *Quake*, however, is one of the few games to allow the querying of players' IP addresses, which may be useful for determining the network topologies and spatial analysis of games. The sourcecode for the game is freely available, so this set of observations may prove useful for future work.

The last pair of observations, O-IV and O-V, come from two sets of servers which Microsoft Research have been running at their site in Cambridge, UK. Using a public list of servers proved to have difficulties, since some of the IP addresses on the list were dynamically allocated (for instance, users running game servers on dial-up machines). It would appear that the master list does not update frequently enough to eliminate these, and so many polls would end up targeting machines which were no longer running the game server. Of the original list of 2193 addresses, we found that 265 of these were never running the server during the course of our polls. The game used in O-V, *Quake III Arena*, does not allow the querying of player duration. For this set of data we assume that each player joined at the time of the poll at which they are first noticed; this figure thus has an inaccuracy of up to two poll periods.

3.1 Summary of observations

We observed a total of 1,757,539 individual sessions (i.e., individual users joining and then leaving a game). Table 2 shows some of the overall aspects of the data. We were interested in examining three specific features: the number of participants in a game, the interarrival time between participants, and how long a player remained in a game.

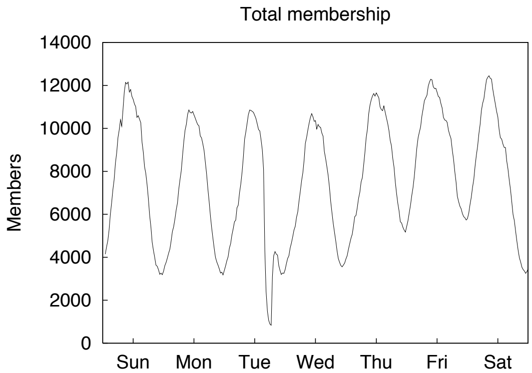
4 Session membership

Figure 3(a) shows the total number of players for all the servers in O-I and O-III to O-V, aggregated to a one-week period. Figure 3(b) shows the number of players for one server from O-IV, again scaled to one week. It can be seen that the number of participants in a game exhibits strong time-of-day effects, peaking in the middle of the day. The strong sinusoidal pattern in the correlograms in Figures 3(c) and 3(d) also indicates seasonal variation.

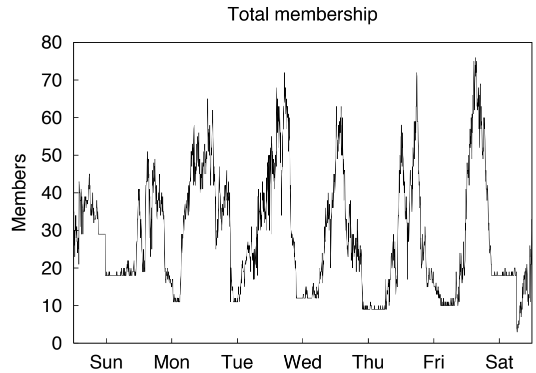
It can be seen from Figure 3(a) that mid-Tuesday is an outlier, with an unusually low number of players. This might have been due, for instance, to a break in network connectivity. This outlier was removed by replacing the data with the average of the samples 24 hours before and 24 hours after.

	Total joins	*Average joins/server/hr*	*Median interarrival (sec)*	*Max interarrival (sec)*	*Median duration (sec)*	*Max duration (sec)*
O-I	1510445	4.65	225	246171	1576	3165999
O-II	69961	27.76	70	17309	1098	66738
O-III	37037	10.01	118	51706	618	410699
O-IV	23559	4.19	115	77843	612	2614737
O-V	5872	1.04	300	76800	901	403201

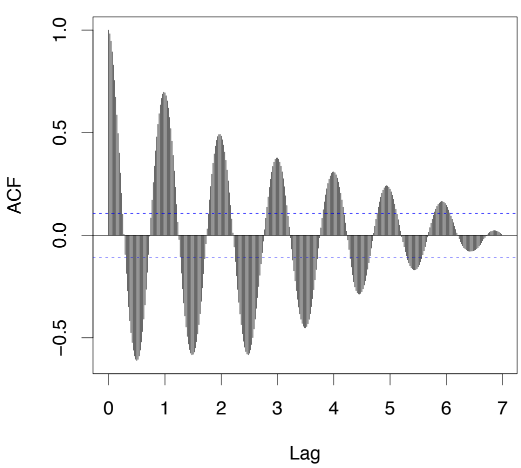
Table 2: Summary of results



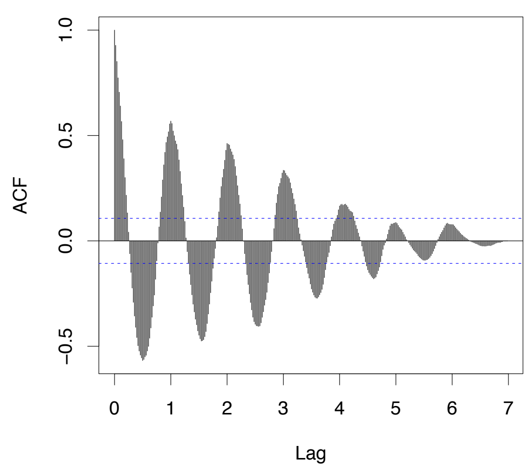
(a) O-I, O-III, O-IV, O-V aggregated



(b) single server from O-IV



(c) ACF of Figure 3(a)



(d) ACF of Figure 3(b)

Figure 3: Number of users

Since the time-of-day effect is so clearly evident, it is possible to do a simple seasonal decomposition by subtracting each observation from the mean value for all the observations taken at that time of day [7]. The results of this are shown in Figure 4, where the higher solid line represents the time-of-day effect, the lower solid line the remainder, and the dashed line the observed data. Three days are higher than the others; these, as one might expect, are Friday to Sunday.

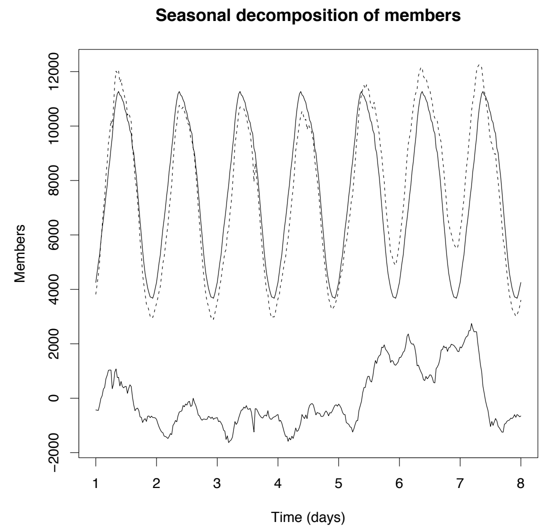


Figure 4: Seasonal decomposition of smoothed membership data

4.1 Network externalities

It is accepted that the value of a group activity to an individual participant may be related to the number of participants in that group. This has been quantified conjecturally by engineers as Metcalfe's Law (the value of a network is proportional to n^2, where n is the number of users [23]), or more recently as the Group-Forming Law (the value of the Internet is proportional to 2^n [28]). Economists, however, generally refer to these effects as "positive consumption" or "network externalities": for example, Katz and Shapiro define network externalities as "products for which the utility that a user derives from consumption of the good increases with the number of other agents consuming the good." [18] Network externalities have most commonly been studied in terms of standardisation and compatibility (e.g., the take-up and acceptance of fax machines [11]), although Henriet and Moulin [16] present a cost allocation scheme for networks where users share costs according to the network externalities that are accrued.

One would expect that multiplayer games would also exhibit network externalities. The purpose of a networked multiplayer game is to participate with other people; if a user wishes to play against electronic opponents there would be less need for the networked aspect of the game (unless, for example, a user wished to play against a far more powerful computer such as the famous chess matches between Kasparov and IBM machines). In general, however, it is reasonable to assume that a given participant in a networked game is taking part because they wish to interact with other remote, human users, and, therefore, that their utility is derived, to some extent, from the existence and number of these other users.

Figure 5 shows the temporal ACF (autocorrelation function) of the corrected data from Figure 4, after removing the time-of-day effect; this shows the degree to which the number of players in a subsequent time period depends on the session membership in the previous period. It can be seen that the level of autocorrelation is high, even for a large number of time periods. Thus, as expected, there appear to be some network externality effects.

Having observed the time-of-day and network externality effects, we analysed the session membership data using time-series analysis. ARIMA (Autoregressive Integrated Moving Average) models, introduced

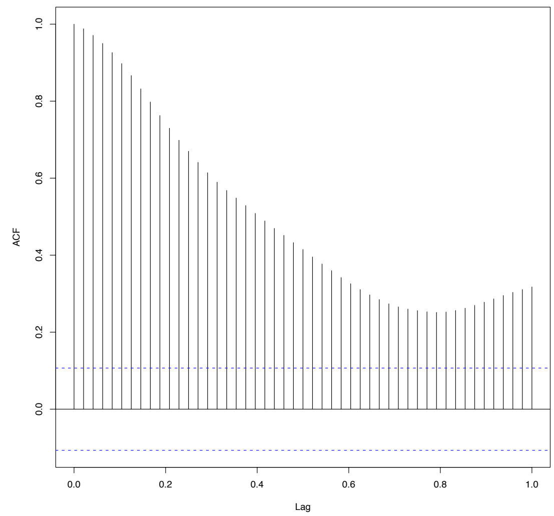


Figure 5: Temporal autocorrelation in number of players

by Box and Jenkins [6] are a popular means of modelling time series data. An autoregressive process is defined as a serially dependent process whereby elements in a time series can be described in terms of previous elements:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + ... + \varepsilon \tag{1}$$

A moving average process is where each element in a time series is affected by past errors, independent of the autoregressive process:

$$X_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \theta_3 \varepsilon_{t-3} - ... \tag{2}$$

An ARIMA model incorporates both the autoregressive and moving average processes. Such models are referred to as ARIMA(p,d,q), where p is the autoregressive parameter, d the number of differencing passes required to make the input series stationary, and q the moving average parameter. If the time series has a seasonal component, additional seasonal parameters are required, and the model is referred to as an ARIMA$(p,d,q) \times (P,D,Q)_s$ model, where P, D and Q represent the ARIMA parameters of the seasonal component, and s is the period of the seasonality.

For the aggregate session membership data from Figure 4, there is little to choose between a $(1,1,1) \times (0,1,1)_{48}$ and a $(2,1,1) \times (0,1,1)_{48}$ model (Figure 6 shows the diagnostic output for the latter). Applying these two models to individual servers' data, however, showed that a $(2,1,1) \times (0,1,1)_{48}$ model is the most appropriate. Figure 7 shows the diagnostics for one server; the Box-Pierce statistic indicates a high goodness of fit.

A $(2,1,1) \times (0,1,1)_{48}$ model incorporates both network externalities, since the autoregressive component means that the number of players up to an hour prior to a player joining has an effect on a player's decision to join, and also includes the time-of-day effect through the seasonal $(0,1,1)_{48}$ component.

Proportional fairness has become a popular metric for allocating bandwidth between flows on a congested link [19]. This relies on the assumption of users having logarithmic utility functions. It is unclear, however, whether this same unicast logarithmic utility function should be assumed for a multicast or multipoint transmission. Chiu [8] shows that proportional fairness may produce an "unfair" outcome in the multicast case, and proposes a weighted proportionally fair solution, where multicast flows receive a bandwidth share weighted according to the aggregate utility of the downstream receivers. Legout *et al.* [20] suggest that this bandwidth share might relate either linearly or logarithmically to the number of downstream receivers. The data presented here supports the latter suggestion, since they imply that it might be more appropriate to assume individual utility functions which incorporate network externalities.

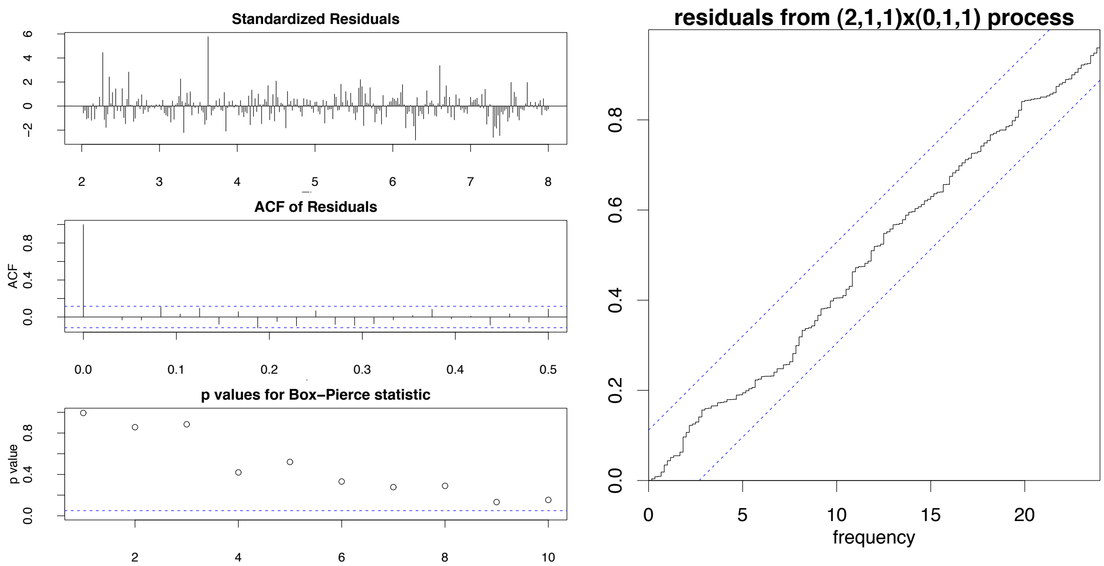


Figure 6: ARIMA diagnostics and cumulative periodogram for $(2,1,1) \times (0,1,1)_{48}$ model on data from Figure 4

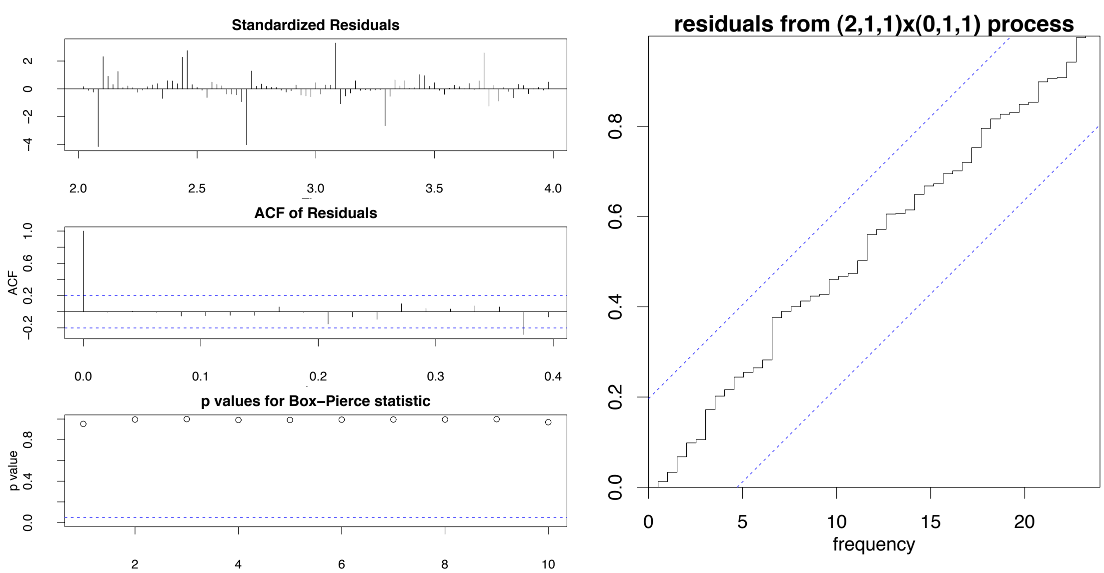


Figure 7: ARIMA diagnostics and cumulative periodogram for $(2,1,1) \times (0,1,1)_{48}$ model for single server

Time-of-day pricing is commonly used for pricing utilities such as electricity and telephone service, and has been proposed as a simple, if suboptimal, method for pricing Internet traffic [21]. The time-of-day effects observed here mean that this might be appropriate on a per-application basis, at least for games. This might also have implications for network provisioning, whereby a network designed for games would want to be able to deal with the peaks.

5 User duration

Game servers tend to run continuously, with users joining and leaving as they wish. As such it is not meaningful to discuss the overall session duration, i.e., a whole game. Instead we examine the duration

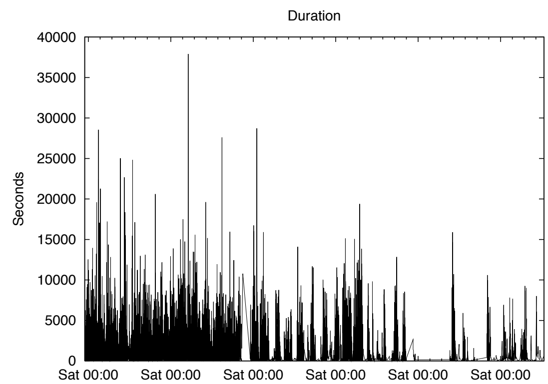


Figure 8: Duration of user's game

of each individual session, i.e., a user's game. Figure 8 shows the duration of users' individual sessions from O-II: it can be seen that these durations vary quite widely, and that many game durations are lower than our polling period of five minutes. This might be due to dropped connections, or users browsing games by starting a session to see what is going on and deciding that a particular game is not to their liking. At the other end of the spectrum, there are several long game durations of over 24 hours. These might be "hardcore" gamers, automated players/bots or users who have mistakenly left their connections active.

In Figure 9 we fit the user duration data for two individual servers to a set of randomly generated exponentially-distributed data. The Quantile-Quantile plots show that this is an appropriate model. This agrees with the findings for multicast sessions in [2]. It is also known that some single-user applications, such as voice telephone calls [4] fit an exponential distribution.

Since we had already observed network externality effects in the number of players, we expected to find a correlation between the duration of a player's session and the number of players in that game; a game with more players might be likely to lead to players enjoying the game more, which should lead to them staying longer. Surprisingly, there appears to be no evidence for this. Figure 10(a) shows a boxplot of the number of players at the start of a player's session against the duration of their session. There does not seem to be a correlation, and the median duration is relatively constant irrespective of the number of players. Comparing the duration to the average number of players over the first hour of a session (Figure 10(b)) showed a slight correlation, but this was insignificant. This might indicate that the absolute number of players in a session is not necessarily a determinant of when a player decides to leave a session; it may be the behaviour or skill of the specific players that is more important, or a completely unrelated factor.

6 Interarrival times

Figure 11 shows the interarrival times between players for one server. As for duration, there is large variation. Unlike the duration data, interarrival times do not appear to fit an exponential distribution, as shown in Figure 12.

Interarrival times between users for single-user applications have been found to fit a Poisson distribution [12, 26]. This is unlikely to be the case for multiuser applications, however, where the presence of other users may alter user behaviour. Borella [5] finds that for games, packet interarrival times are highly correlated. Figure 13 shows that this is also true for player interarrivals; there is significant autocorrelation at short lags, which implies that the arrival of some users will lead to others arriving. Thus, the interarrivals do not fit the independent arrivals of the Poisson distribution.

Heavy-tailed distributions have been observed for Internet usage behaviour, for example in World Wide Web usage [9] and aggregate Ethernet traffic [30]. One method for visualising a heavy-tailed distribution is a log-log complementary distribution (LLCD) plot, where the complementary cumulative distribution is plotted on logarithmic axes. Linear behaviour in an LLCD plot indicates a heavy-tailed distribution. Figure 14(a) shows such a plot for the interarrival times, and linear behaviour can be observed for the

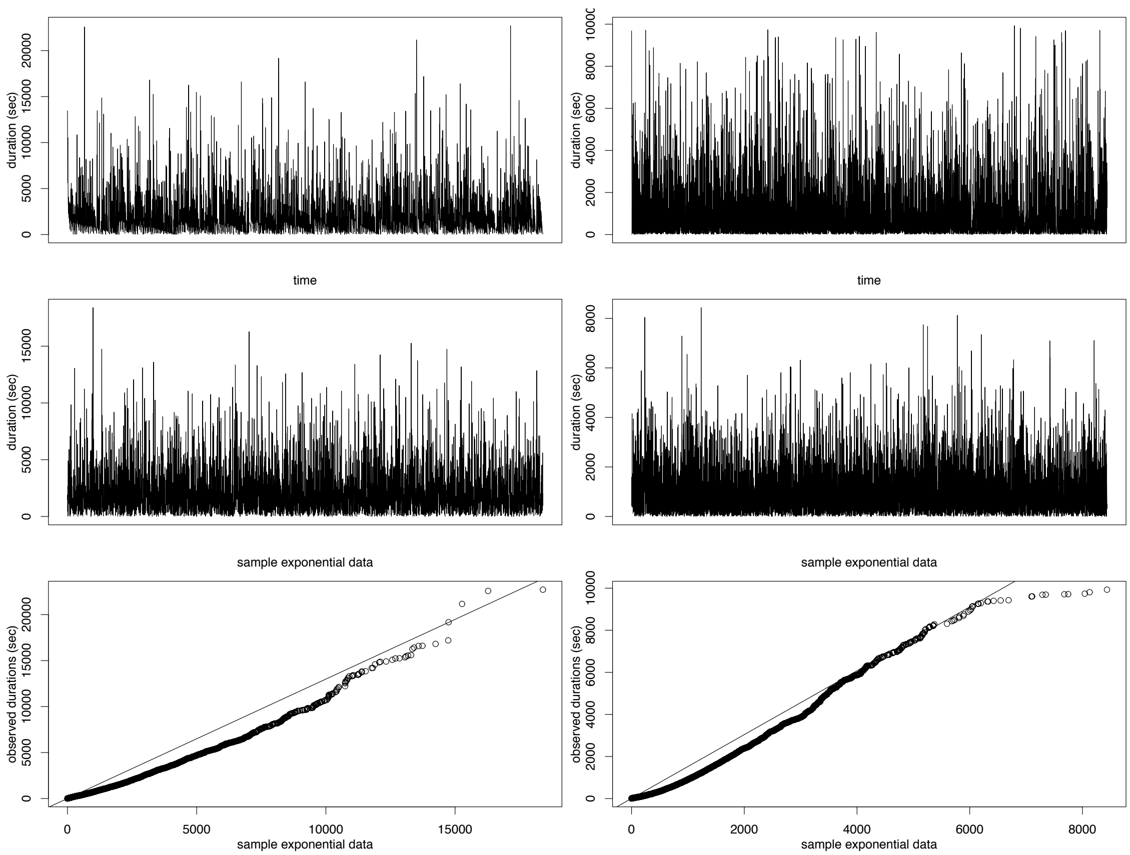


Figure 9: Fitting an exponential distribution to user duration data

larger observations (Figure 14(b)).

A more rigorous test for heavy-tailed distributions is the Hill estimator [17]. A distribution of variable X is heavy-tailed if

$$P[X > x] \sim x^{-\alpha}, \text{as } x \to \infty,\ 0 < \alpha < 2. \tag{3}$$

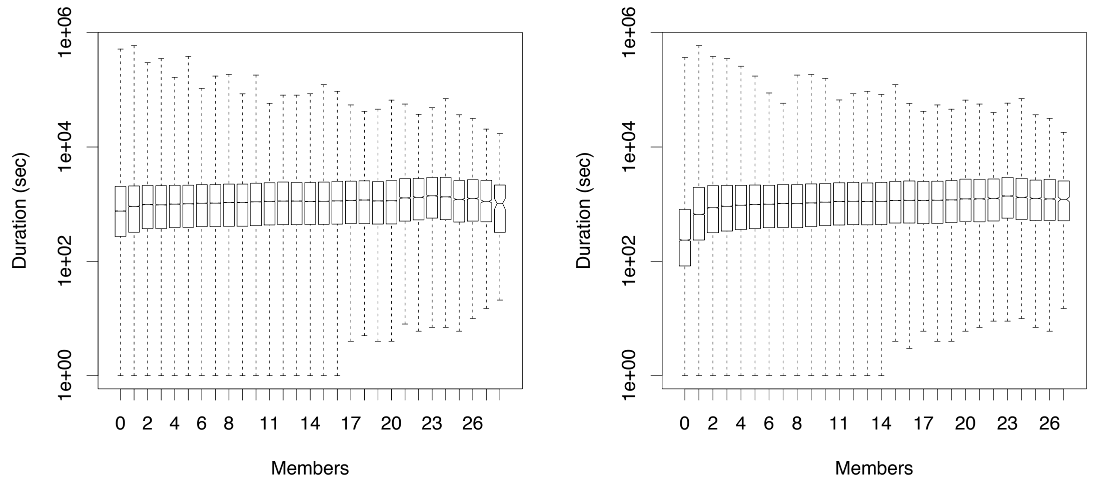
The Hill estimator can be used to calculate α

$$\hat{\alpha}_n = \left(1/k \sum_{i=0}^{i=k-1} (logX_{(n-i)} - logX_{(n-k)}) \right)^{-1} \tag{4}$$

where n is the number of the observations, and k indicates how many of the largest observations have been used to calculate $\hat{\alpha}_n$. Figure 15 shows that $\hat{\alpha}$ is approximately 1.15.

Comparing the interarrival times to the number of players in a session shows some evidence of an inversely proportional relationship (Figure 16); as the number of players in a session increases, the interarrival times decrease. This supports the hypothesis that the number of players is a determinant in other players' decisions to join a session.

The high variation in user duration and interarrival times have several implications for price stability and provisioning if the members of a multicast group are to share the overall cost of a session amongst themselves. The autocorrelation in the number of players and interarrival times means that if the users are sharing the costs of a session, this cost will snowball; new users joining will be followed by other users joining (and users will join faster as the number of users increases), leading to rapid decreases in the cost per user, and vice versa for when users leave. This could be rectified, for example, by only changing the price for each user on a periodic basis rather than with each join or leave. The autocorrelation seems to exist for large lags, however, which means that the periods of price reevaluation would also have to be large, and this could impede the efficiency of any pricing scheme.



(a) Number of players at start of session

(b) Average number of players over first hour of session

Figure 10: Number of players versus duration of session

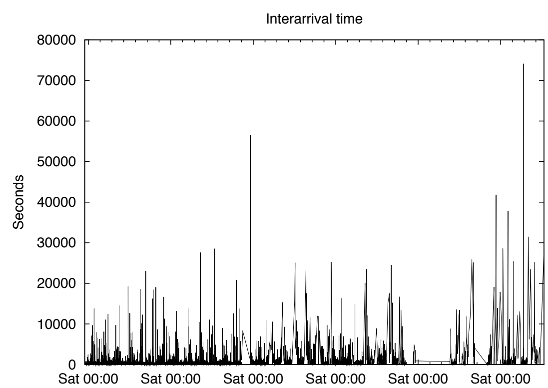


Figure 11: Interarrival times

Cost sharing may also change the behaviour of players, since we have observed that the number of players in a session is not a large factor in the utility received from a game. Once network pricing or QoS are a feature of networks, then users will need to choose between network flows depending on the value that they receive from each application; in other words, they will attempt to maximise their utility given their individual budgets. The price of a session thus becomes a factor in user behaviour, and if the cost of a session is shared amongst session members, then the number of players in a session will become a factor, since it will be a determinant in the price of the session. Additionally, the number of users in a session might affect the QoS, which would become a further factor for users' behaviour.

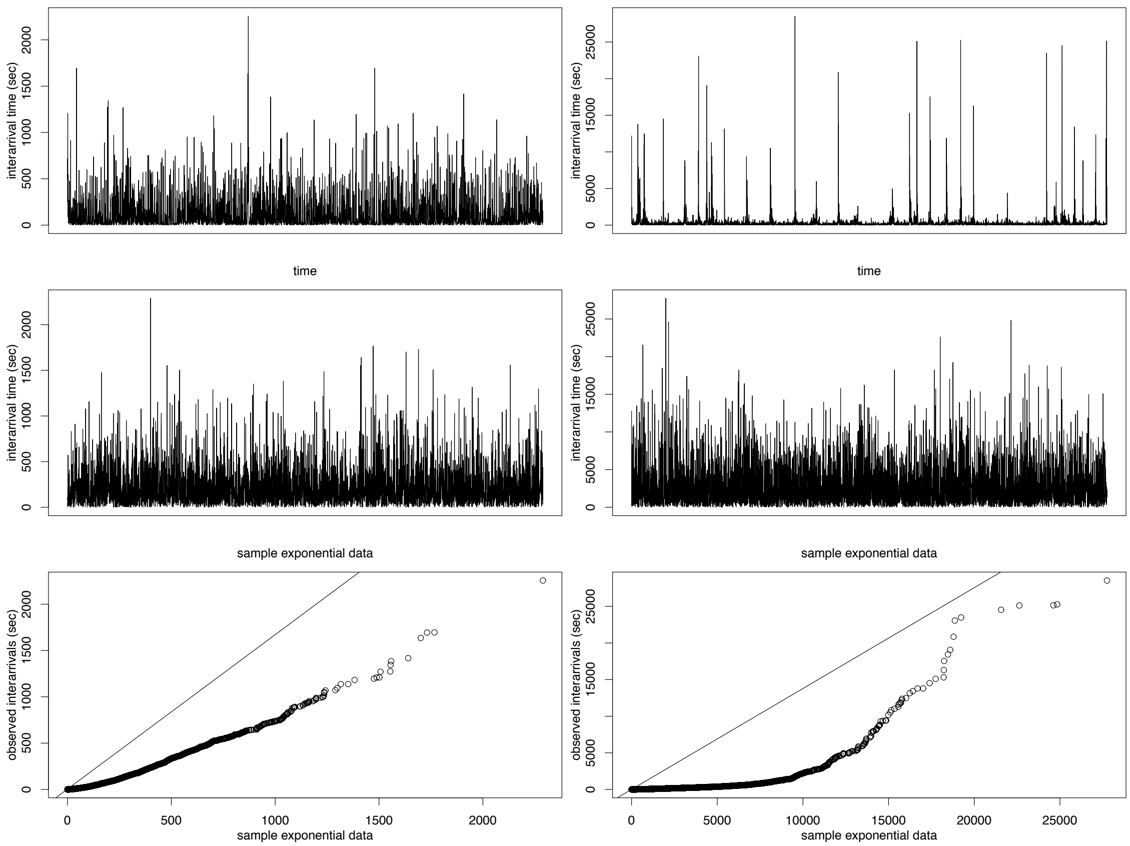


Figure 12: Fitting an exponential distribution to interarrival times

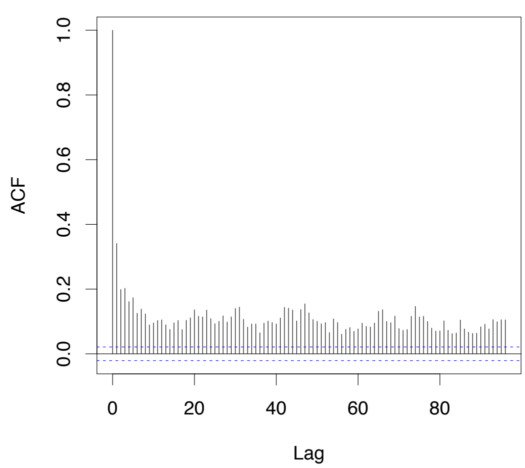
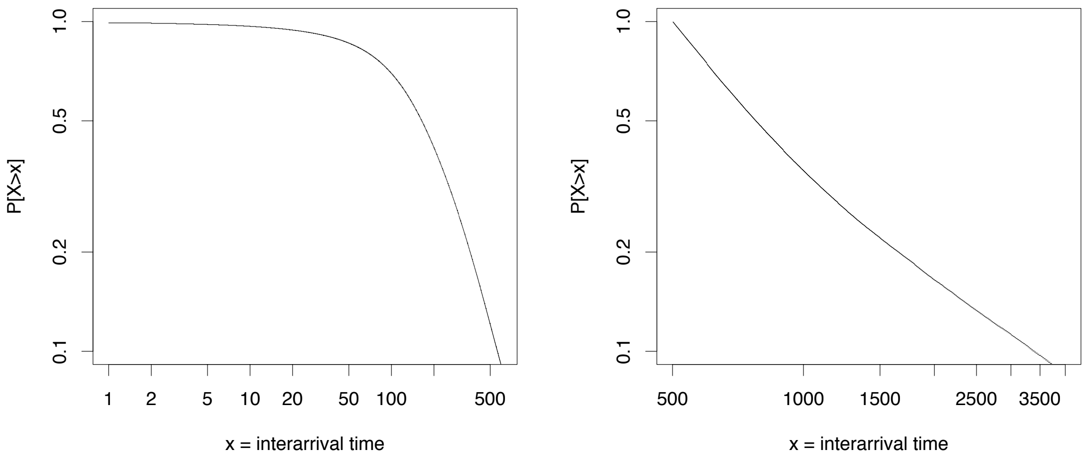


Figure 13: ACF of interarrival times

7 Conclusions and future work

There has been little study of session-level user behaviour in large-scale multiple-source scenarios. In this paper we have presented statistical analysis of several session-level traces of popular multiplayer net-



(a) full data set (b) upper tail

Figure 14: Log-log complementary plots of interarrival times

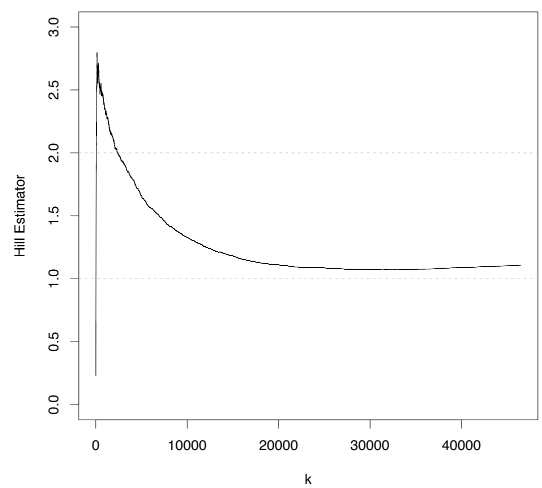


Figure 15: Hill estimator for interarrival times

worked games. We have found that the number of players exhibits strong time-of-day and network externality effects, and we have fitted an appropriate ARIMA model. Players' duration times fit an exponential distribution, while interarrival times fit a heavy-tailed distribution. The number of players in a session appears to have a greater effect on players' decisions to join a session rather than leave. In many respects we have observed similar behaviour to that seen for multicast applications, despite the unicast nature of these games. This implies that in the absence of appropriate multicast data, unicast multipoint applications are an appropriate substitute. We have discussed how these results could impact potential multicast pricing

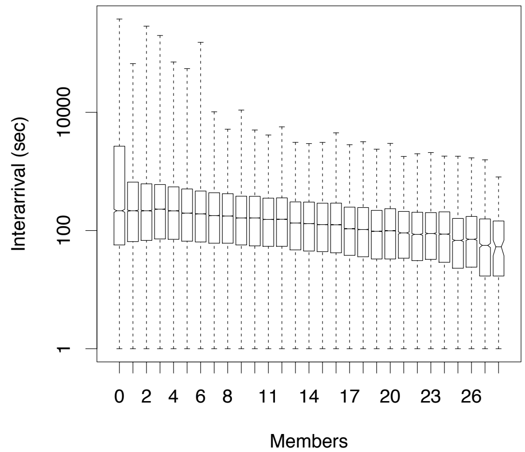


Figure 16: Number of players versus interarrival time

policies and network provisioning.

Networking research into multiplayer games is still at an early stage, and there is much future work that needs to be carried out. Understanding user behaviour is but one stage in creating appropriate pricing policies. It does not, for example, help us explain what users desire or require. Future work will investigate the QoS requirements for networked games, in particular, concentrating on how the requirements change depending on the composition of a session and group.

We also intend to look at packet-level traffic statistics. Some of the results presented here have been similar to those of previous packet-level games studies, and it will be interesting to conduct simultaneous analysis at the packet and session levels, to determine whether there is any relationship between the behaviour at both levels. This is important, for example, if QoS provisioning is to take place through congestion pricing, i.e., charging users for the network congestion that they cause. Unfortunately, most of the game server operators and ISPs that we have spoken to do not log many of the appropriate statistics. Thus, we are now running our own games servers in order to collect packet-level data. Running our own servers presents many opportunities for further work. Since we can log the exact times when players connect and leave to the server, we can better estimate the inaccuracy of the polling method that we have used here. We are also using these games servers for experimental rather than correlational study; for example changing such variables as network delay in order to determine the effects on user behaviour.

The study presented here has only looked at one type of game, the FPS. Although we examine two of the most popular games of this genre, it is not necessarily true that these results will hold for other FPS games and this should be further examined. Moreover, other types of games, for instance MMORPGs (Massively Multiplayer Online Role-Playing Games) such as Everquest, are likely to exhibit different user behaviour, since the MMORPG can be slightly slower paced, and can involve thousands of users connecting to a single server, rather than the large number of small groups in FPS games.

8 Acknowledgements

Thanks to Brendan Murphy at Microsoft Research Cambridge UK for allowing us to poll their games servers.

References

[1] 3Com. 3Com scores with modem geared towards serious gamers, Oct. 25, 1999. `http://www.3com.com/corpinfo/en_US/pressbox/press_release.jsp?INFO_ID=2001253`.

[2] K. C. Almeroth and M. H. Ammar. Collecting and modeling the join/leave behavior of multicast group members in the MBone. In *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing (HPDC-5)*, pages 209–216, Syracuse, NY, Aug. 1996.

[3] R. A. Bangun, E. Dutkiewicz, and G. Anido. An analysis of multi-player network games traffic. In *Proceedings of 1999 International Workshop on Multimedia Signal Processing*, pages 3–8, Copenhagen, Denmark, Sept. 1999.

[4] V. A. Bolotin. Modeling call holding time distributions for CCS network design and performance analysis. *IEEE Journal of Selected Areas In Communications*, 12(3):433–438, Apr. 1994.

[5] M. S. Borella. Source models of network game traffic. *Computer Communications*, 23(4):403–410, Feb. 15, 2000.

[6] G. E. Box and G. M. Jenkins. *Time series analysis: forecasting and control*. McGraw-Hill, London, UK, 1970.

[7] C. Chatfield. *The Analysis of Time Series*. Chapman & Hall, London, UK, fifth edition, 1996.

[8] D. M. Chiu. Some observations on fairness of bandwidth sharing. Technical Report TR-99-78, Sun Microsystems Research, Palo Alto, CA, Sept. 1999.

[9] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, Dec. 1997.

[10] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14(1):78–88, Jan. 2000.

[11] N. Economides and C. Himmelberg. Critical mass and network evolution in telecommunications. In *Proceedings of the 22nd Telecommunications Policy Research Conference*, Solomon's Island, MD, Oct. 1994.

[12] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz. The changing nature of network traffic: Scaling phenomena. *Computer Communication Review*, 28(2):5–29, Apr. 1998.

[13] GameSpy. `http://www.gamespy3d.com`.

[14] Half-Life. `http://www.sierrastudios.com/games/half-life/`.

[15] T. N. H. Henderson and S. N. Bhatti. Protocol-independent multicast pricing. In *Proceedings of the 10th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 11–17, Chapel Hill, NC, June 2000.

[16] D. Henriet and H. Moulin. Traffic-based cost allocation in a network. *RAND Journal of Economics*, 27(2):332–345, Summr 1996.

[17] B. M. Hill. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, 3(5):1163–1174, Sept. 1975.

[18] M. L. Katz and C. Shapiro. Network externalities, competition, and compatibility. *American Economic Review*, 75(3):424–440, June 1985.

[19] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, Jan./Feb., 1997.

[20] A. Legout, J. Nonnenmacher, and E. Biersack. Bandwidth allocation policies for unicast and multicast flows. In *Proceedings of the 18th IEEE Conference on Computer Communications (INFOCOM)*, pages 254–261, New York, NY, Mar. 1999.

[21] J. K. MacKie-Mason and H. R. Varian. Some economics of the Internet. In W. Sichel and D. L. Alexander, editors, *Networks, Infrastructure and the New Task for Regulation*. University of Michigan Press, 1996.

[22] S. McCreary and K. Claffy. Trends in wide area IP traffic patterns: A view from Ames Internet Exchange. In *ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, Monterey, CA, Sept. 2000.

[23] R. M. Metcalfe. The Internet after the fad. In *The 1996 Monticello Lectures*, Monticello, VA, May 1996.

[24] T. Monk and K. Claffy. Internet data acquisition and analysis: Status and next steps. In *Proceedings of the 7th Internet Society Conference (INET)*, Kuala Lumpur, Malaysia, June 1997.

[25] A. Patrizio. Coming soon: Pay-per-game. *Wired News*, Oct. 20, 2000. `http://www.wired.com/news/culture/0,1284,39505,00.html`.

[26] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[27] QStat. `http://www.qstat.org`.

[28] D. P. Reed. Weapon of math destruction. *Context Magazine*, Spring 1999.

[29] Urban Mercenary. `http://www.urbanmercenary.com`.

[30] W. Willinger, V. Paxson, and M. S. Taqqu. Self-similarity and heavy tails: Structural modeling of network traffic. In R. J. Adler, R. E. Feldman, and M. S. Taqqu, editors, *A Practical Guide to Heavy Tails — Statistical Techniques and Applications*, pages 27–53. Birkhäuser, Boston, MA, 1998.