

27-3-2000

## **Modelling multi-player games traffic**

R. A. Bangun

*University of Wollongong, [uow@bangun.edu.au](mailto:uow@bangun.edu.au)*

E. Dutkiewicz

*University of Wollongong, [eryk@uow.edu.au](mailto:eryk@uow.edu.au)*

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

---

### **Recommended Citation**

Bangun, R. A. and Dutkiewicz, E.: Modelling multi-player games traffic 2000.  
<https://ro.uow.edu.au/infopapers/204>

---

## Modelling multi-player games traffic

### Abstract

The growing popularity of delay-sensitive multi-player games over the Internet is creating a need for proper characterisation of such traffic. We are currently developing source models for game traffic, on a per-player basis, with the intended use being to scale the models to simulate scenarios involving large numbers of players, which will enable us to observe the effects of game traffic on the network. We outline the procedure used, and we show that for certain cases, relatively simple source models are able to provide sufficiently accurate results (in terms of mean delay and mean buffer occupancy).

### Keywords

Internet, computer games, digital simulation, multi-access systems, telecommunication computing, telecommunication traffic

### Disciplines

Physical Sciences and Mathematics

### Publication Details

This paper originally appeared as: Bangun, RA & Dutkiewicz, E, Modelling multi-player games traffic, Proceedings. International Conference on Information Technology: Coding and Computing, 27-29 March 2000, 228-233. Copyright IEEE 2000.

# Modelling Multi-Player Games Traffic

Ricky A. Bangun and E. Dutkiewicz

Telecommunications and Information Technology Research Institute  
School of Electrical, Computer and Telecommunications Engineering  
University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia  
Email: [ricky@snrc.uow.edu.au](mailto:ricky@snrc.uow.edu.au)

## Abstract

*The growing popularity of delay-sensitive multi-player games over the Internet is creating a need for proper characterisation of such traffic. We are currently developing source models for game traffic, on a per-player basis, with the intended use being to scale the models to simulate scenarios involving large numbers of players, which will enable us to observe the effects of game traffic on the network. In this paper, we outline the procedure that we use, and we show that for certain cases, relatively simple source models are able to provide sufficiently accurate results (in terms of mean delay and mean buffer occupancy).*

## 1. Introduction

Over the last few years, many new Internet applications have emerged, offering increasing degrees of interactivity, either between the human and the application, or between two or more human users. An example of the latter are multi-player games, which range from network-delay-insensitive games such as online chess, to the network-delay-sensitive games (better known as “fast-twitch” games) which require speed and skill from the human player, e.g. the well-known *Quake* series from *id Software* [1]. Multi-player games that are insensitive to network delays are not dissimilar in nature to applications such as e-mail. In other words, they are easily supported by current networks, and therefore present no immediate concern to us. Instead, our research focuses on the delay-sensitive multi-player games. Our paper discusses our efforts into modelling the traffic generated by such applications. In particular, we show that, in certain cases, relatively simple models can be used for the traffic generated by a player in a multi-player game session.

Multi-player games are generally more interesting than their single-player counterpart, due to the unpredictability of the human players, and the essentially inexhaustible pool of potential players. The commercial appeal of

multi-player games have been proven by the increasingly large base of players. For example, *Mplayer.com*, the top-rated Internet games service provider, has seen a sharp rise in membership numbers in less than 3 years [2] (see Figure 1). The game console industry has also recognised the rapid growth of multi-player games, evidenced by the fact that (Internet) connectivity is the main selling point of the newest batch of game consoles such as *Dreamcast* (from *Sega Corp.*) and the upcoming *PlayStation II* (from *Sony Corp.*).

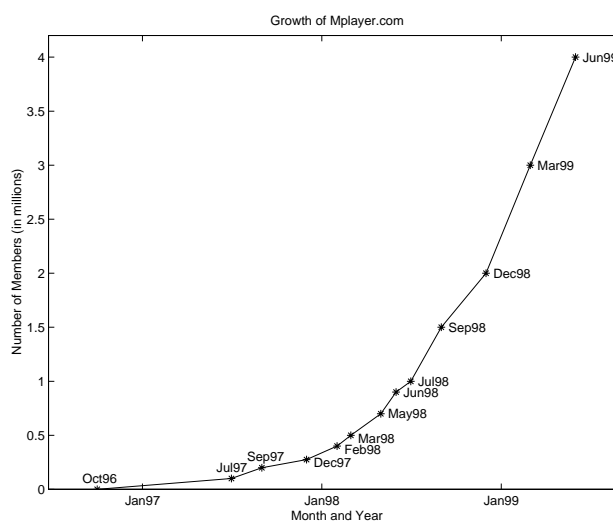


Figure 1. Membership numbers for Mplayer.com

Increase in the usage of multi-player games over the Internet points to a corresponding increase in the amount of Internet traffic attributed to multi-player games. This situation brings about the question regarding the way in which large loads of game traffic would affect current and future networks. In order to properly address this, it is imperative to understand the characteristics of the traffic generated by multi-player games. At the present time, these characteristics are poorly understood, as can be seen by the lack of papers addressing this issue [3], [4].

Our research is aimed at developing game traffic models that can be used to assess the impact of large loads of game traffic on current and future networks. Our approach is to develop per-player source models based on real multi-player game traffic traces. These source models can then be scaled to reflect the desired number of players, and – together with a suitable network model – can be used to study the impact of game traffic on networks.

## 2. Data collection

For our modelling purposes, we have collected network traffic traces from a local area network which was used for semi-regular multi-player game sessions. The data collection setup consists of 16 computers connected to a dumb hub, with one of the computers acting as the traffic monitor, as shown in Figure 2. This computer was running *FreeBSD version 3.1-Release* [5], and collecting game packet data using *tcpdump* [6]. Packet data that was not game-related was filtered out from the traces, and the remaining data was cleaned and sorted on a per-player basis. A database consisting of several popular game titles, with varying number of players, was compiled.

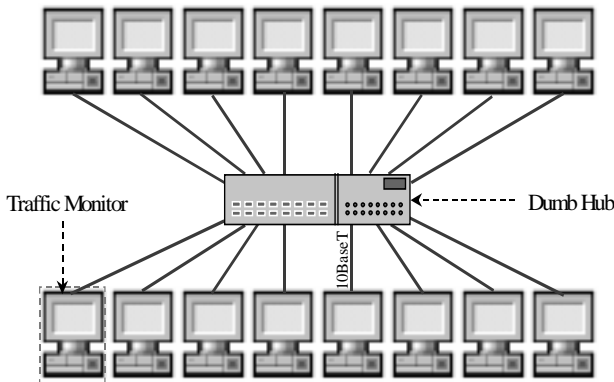


Figure 2. Data collection setup

From this point forward, we will focus on an example trace, which is a combined trace from 7 players of the game *Starsiege Tribes*, collected on 14<sup>th</sup> March 1999. *Starsiege Tribes* (or simply *Tribes*) is a product of *Sierra Online Entertainment*, and was popular in 1999. The game is designed using a client-server based communications model, in that the players' machines communicate with each other via a central server. Packet transmission is done using the User Datagram Protocol / Internet Protocol (UDP/IP). Due to limited space in this paper, we will not discuss the details of this game any further. Note that since we are interested in obtaining per-player source models, in this paper we concentrate solely on the traffic generated by the clients of this game.

## 3. The typical player

In any multi-player game, a wide assortment of human users would be involved, bringing with them a variety of playing tactics and strategies. Examples of what perhaps may be the two extreme cases are the 'constantly-moving-and-attacking' approach, and the 'ambush-an-attack' approach. In the former case, the player moves around the virtual world of the game as fast as s/he can, possible generating traffic more often than a player in the latter case, who would typically just wait at particular places in the virtual world and ambush other players who wander into that location. The different payload size distributions attributed to different players is illustrated in Figure 3 below. The figure is in fact a collection of histograms for each player, with bin sizes of 3 bytes. But for clarity purposes, we simply plotted the midpoints of the histogram bars, connected together using lines.

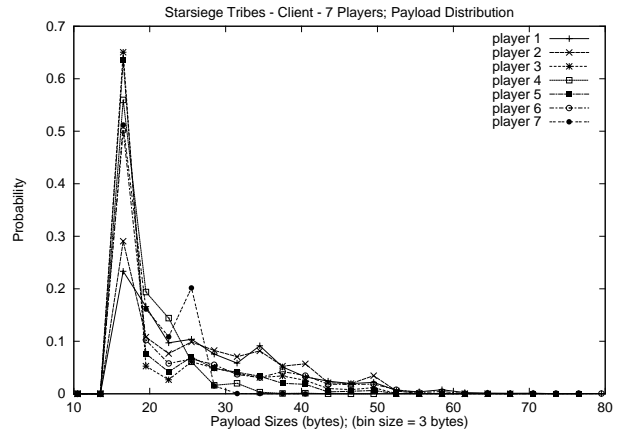


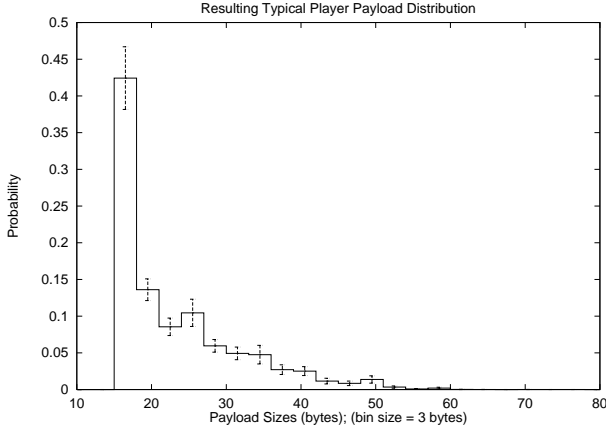
Figure 3. Payload distribution for each player

We can see from Figure 3 that the general trend is similar for all the players, but the individual distributions are different. This makes it difficult to randomly select a particular player which will be used as a basis for the source model. We have chosen to instead develop what we call the *typical* player.

The distribution for the typical player is simply the average of the distribution for all the individual players. However, we do not mix player distributions indiscriminately. Rather, we only use distributions belonging to players who were involved in game sessions of the *same game* and the *same number of players*. For example, we would have a typical player for Tribes with six players, and another typical player for Tribes with seven players, etc. Where data is available, more players tend to lead to better representations of the typical player.

In Figure 4 we show the result of developing a typical player for the game Tribes with 7 players. This typical player is a result of averaging the payload distributions of 35 players (from 5 separate game sessions of Tribes, each

consisting of 7 clients and one server – we only use the clients’ data). It is the empirical distributions of the typical player that we then use to develop our source models.



**Figure 4. Payload distribution of typical player**

#### 4. Distribution fitting process

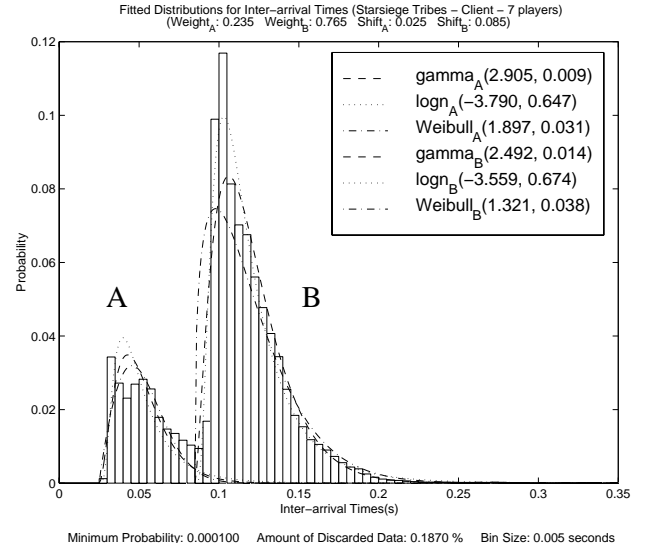
For the purpose of modelling the per-player traffic traces, we analysed and fitted probability distributions to payload size and inter-arrival time distributions of typical players. Note that the payload size does not include any packet headers; rather, it is simply the size of the data transmitted by the player. We find that the general shape of the distributions for these statistics is related to the number of players involved in the game session [7]. The procedure for fitting the distributions is as follows:

1. Create a short list of possible distributions, based solely on the fact that the shape of the distributions follow the general trend of the empirical distribution.
2. Cull extreme outliers from the distribution. This step was taken since the behaviour of the tail of our distribution is not so important in our case, and we have observed that culling extreme outliers results in much better fitting distributions. We should also point out that the outliers in our data are usually attributed to one or two packets (out of a few hundred thousand) which have larger payloads and/or inter-arrival times. These outliers are not part of the tail of the distribution, but are instead anomalous occurrences.
3. Compute the necessary positive shift using a simple formula [8]. This step was required, as in many cases a shifted distribution provided a much better fit to the empirical distribution.
4. Compute the values of the required parameters for the distributions using the *Maximum Likelihood Estimator* (MLE) method.

In terms of the distribution fitting process, the payload size and inter-arrival time statistics for the various games for which we have collected data revealed three distinct cases, as follows:

- The empirical distribution was dominated by two or three closely-spaced discrete values, accounting for over 90% of the data. For this case, we approximate the empirical distribution with a single value which is the average of the two or three dominant values.
- The empirical distribution was a simple uni-modal distribution, for which the standard technique as described above was used.
- The empirical distribution was bi-modal, in which case the empirical distribution was fitted using split distributions. Essentially this means two separate distributions, with appropriate weightings; the sum of both distributions giving the composite distribution. A concise description of this is given in Equation 1.

$$f_{comp} = \omega_A * f_A + \omega_B * f_B, \quad \text{and } \omega_A + \omega_B = 1 \quad (1)$$



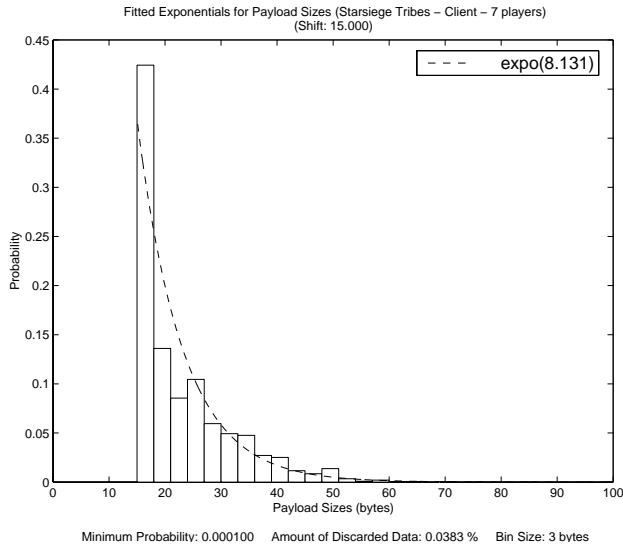
**Figure 5. Example of split distributions**

In the case of Tribes client traffic, the payload size distribution is uni-modal, while the inter-arrival time distribution is bi-modal. An example of the use of split distributions is shown in Figure 5. For example, if we decide that the best fitting distribution for section A (left hand side) in Figure 5 is a lognormal distribution ( $\log n_A$ ), and for section B (right hand side) it is another lognormal distribution ( $\log n_B$ ), then the composite distribution would be:  $0.235 * \log n_A + 0.765 * \log n_B$ . The amount of data that was culled (i.e. the extreme outliers), is also shown in the graph, and in this case it is 0.187 % of the total data, which is a negligible amount.

## 5. Poissonian approximation

During the course of our study, we have found that traffic generated by a player in a game has significant autocorrelation [7], a result also found by Borella [4] who used the game Quake. It is also likely that the traffic generated by players in the same game session would have high cross-correlation. Unfortunately, models that incorporate correlation tend to be more complicated than models that assume data independence.

In order to develop source models that are as simple as possible, we have decided to at first ignore the correlations in the traffic, and apply the assumption of data independence. We have also observed that, although not necessarily providing the best fit to the empirical distributions, in the case of Tribes clients, the exponential distribution (and split distributions consisting of an exponential distribution for each section), provide a reasonably good fit, as shown in Figure 6 (for payload sizes) and Figure 7 (for inter-arrival times).

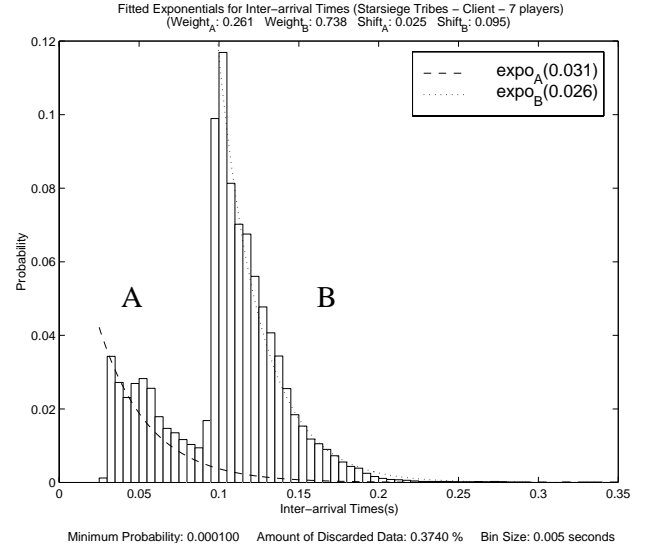


**Figure 6. Fitted exponential distribution for payload sizes**

Essentially, we have assumed that the traffic is Poissonian, and indeed, our results in the following sections seem to indicate that this approximation still provides sufficiently accurate results (in terms of mean delay and mean buffer occupancy for a single server queue). The Poissonian approximation is useful, as source models which are Poissonian are analytically tractable.

Using this approximation, the source model that we obtained for the case of Tribes clients with 7 players turned out to be as follows. Payload sizes are fitted to a shifted exponential distribution, with (unshifted) mean of 8 bytes, and a positive shift of 15 bytes. The inter-arrival

times are fitted using a split distribution, with section A being a shifted exponential distribution with (unshifted) mean of 31 ms, positive shift of 25 ms, and weight of 0.262; while section B being a shifted exponential distribution with (unshifted) mean of 26 ms, positive shift of 95 ms, and weight of 0.738.



**Figure 7. Fitted exponential distributions for inter-arrival times**

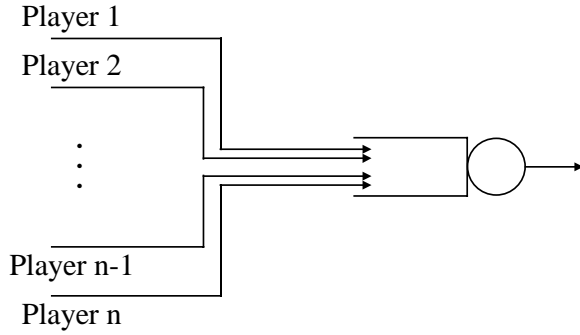
## 6. Simulation setup

Generally, developing analytical traffic models involves an iterative process of testing and fine-tuning the model (and perhaps introducing more complexity into the model), in order to obtain the desired level of accuracy. For the purpose of fine-tuning our models, we need a set of results that can be used as a reference. In other words, a set of results to which we can strive to match by fine-tuning and/or adding more details to the model.

To generate this set of reference results, we used trace-driven simulations of a single FIFO queue with infinite buffer, as shown in Figure 8. The input to the simulation is simply the trace from a Tribes game session with 7 players. The results of interest were mean delay (this includes the packet's service time) and mean buffer occupancy (this includes the size of the packet currently in service) over a range of utilisations. Mean delay is particularly important, as excessive latency has been widely noted as the main factor that affects the playability of fast-twitch multi-player games [9].

This single queue can be thought of as an idealised version of an aggregation point for game traffic in a network. At the present time, an example of such aggregation points is the router at the server complex of a game service provider's site. We have added a fixed

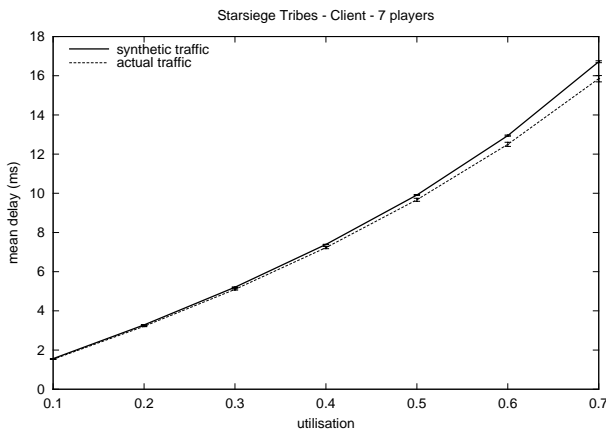
header of 40 bytes to the payloads of the trace before inserting the packet into the queue. This header represents the total UDP/IP header, and is added to create a more realistic scenario.



**Figure 8. Simulation setup**

## 7. Results

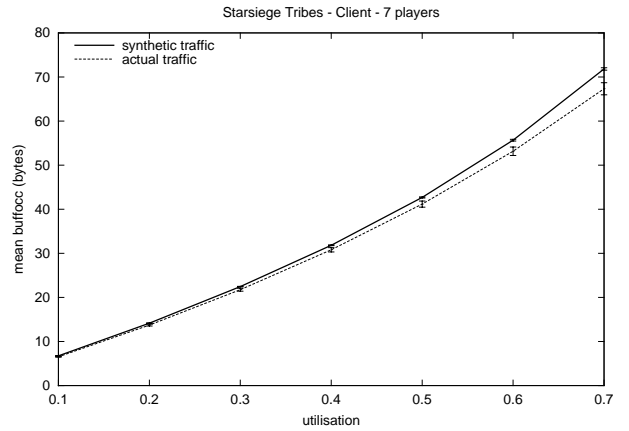
To check the accuracy of our source models, we have generated synthetic game traffic which follows the fitted exponential distributions described in the previous section. We generated seven different traces to represent seven synthetically generated player's traffic, each trace having the same total number of packets. (Recall that we are comparing these results against those obtained using an actual trace from seven Tribes clients.) The synthetically generated traces are therefore independent of each other, and no particular correlation was deliberately introduced. The results of the simulation, using the actual traces and the synthetically generated traces are shown in Figure 9 and Figure 10.



**Figure 9. Mean delay results**

The results that we have shown are for mean delay (including the packet's own service time) and mean buffer

occupancy (including the size of the packet in service). The utilisation level of the system was varied by varying the service rate of the queue's server.



**Figure 10. Mean buffer occupancy results**

As we can see from the figures, the results obtained using the synthetically generated traces closely match the results obtained using the actual traffic traces, up to utilisations of 0.7. After this, the results diverge; however, most networks do not operate at such high utilisations. In both cases, the synthetically generated traces provided results which are larger than those due to the actual traces. In other words, the synthetically generated traces produced conservative estimates of the mean delay and mean buffer occupancy for the case of a single-server queue.

## 8. Conclusions and future work

In summary, this paper describes our efforts into developing sufficiently accurate game source models. We find that in certain cases, relatively simple models can provide sufficient accuracy for results such as mean delay and mean buffer occupancy. Such models may then be scaled to simulate large loads of game traffic. This, in turn, can be used to assist in the study of the impact of (large loads of) multi-player game traffic on current and future networks.

Our immediate future work is to test the applicability of the framework described in this paper to traffic traces from other games.

## 9. References

- [1] id Software website at <http://www.idsoftware.com>
- [2] Various press releases at <http://www.mplayer.com>
- [3] L. Gautier and C. Diot, "MiMaze, a Multiuser Game on the Internet", INRIA research report reference: RR-3248,

September 1997. Available from:  
<http://www.inria.fr/RRRT/RR-3248.html>

- [4] M. S. Borella, "Source Models of Network Game Traffic", in *Proc. of Networld+Interop '99 Engineer's Conference*, May 1999.
- [5] FreeBSD Operating System. Available from  
<http://www.freebsd.org>
- [6] V. Jacobson *et al.*, tcpdump. Available via anonymous ftp from: <ftp://ftp.ee.lbl.gov/tcpdump.tar.Z>, June 1997.
- [7] R. A. Bangun, E. Dutkiewicz, G. J. Anido, "An Analysis of Multi-Player Network Games Traffic", in *Proc. of IEEE MMSP '99*, Copenhagen, Denmark, September 1999.
- [8] M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 2<sup>nd</sup> ed., ch. 6, p. 401, McGraw-Hill, 1991.
- [9] S. Cheshire, "Latency and the Quest for Interactivity", whitepaper, November 1996. Available from:  
<http://ResComp.Stanford.EDU/~cheshire>.