Source Models of Network Game Traffic

Michael S. Borella
3Com Corp.
1800 West Central Rd
Mount Prospect, Illinois 60056

Tel: (847) 342-6093
Fax: (847) 342-6732
Email: mike_borella@3com.com

Abstract

We study the traffic generated by sessions of a popular multi-player network game. Our analysis indicates that empirical game traffic can be characterized well by certain analytical models. While clients and servers, as well as hosts with different configurations, produce different models, all models from the game are well modeled by the same families of distributions. We find that some data sets are best modeled with *split distributions*; that is, one portion of the data is well-modeled with one particular distribution, and the rest of the data with another. We describe how our models can be simulated and discuss how host processing speed influences packet interarrival distributions. As Internet gaming becomes more popular, we expect that our models will be useful for testing hardware and protocols that support gaming.

1 Introduction

Interactive, multi-player network games are becoming more common. The most popular of these games sell millions of copies. Most multi-player games support network play over a LAN or the Internet. When playing a game over the Internet, most players will log on to the network from home using a dial-up PPP connection via a modem. The high latencies that are common on the Internet (typically 50-150 ms roundtrip) as well as the even higher latencies that modem connections exhibit (typically 150-400 ms roundtrip) results in a large user base that is very concerned with network delay. Game players refer to these delays as "lag," [LC] due to the deleterious visual impact that it has on their games.

Interactive gaming is *extremely* delay-sensitive. It is well known that users of interactive voice or video applications (such as Internet telephony or video-conferencing) require roundtrip delays of less than about 300 ms [G114]. However, very few individuals using telephony can tell the difference between 50 ms and 150 ms of roundtrip delay. Game players have found that the difference between 50 ms and 150 ms of delay can determine who wins or loses a game.

Internet service providers (ISPs) are becoming more concerned with lag. In that last 1-2 years, a number of ISPs have been founded with a charter of primarily supporting a high-quality gaming environment for their users. This has led all ISPs to examine and evaluate new and emerging network components, such as modem servers and routers, to ensure that they provide reasonably low delays. It is likely that low-latency optimization will become a primary design goal of network hardware manufacturers in the near future.

In order to properly design and evaluate hardware for a low-latency gaming environment, we require accurate source models of network game traffic. The academic and trade literature has almost completely ignored gaming until very recently. In particular, we would like to find *invariants* of gaming traffic; i.e., statistical characteristics of how a gaming host generates network traffic that can be parameterized for analytical models and simulation. Ideally, such models will be simple and parsimonious. Currently, it is not clear that all game traffic will share the same or similar characteristics. Based on gaming architecture, design, and human factors, there are a number of reasons[1] why game traffic models and parameters may vary significantly:

- *Connectivity:* While most games can be played over a modem or a LAN, clever game designers may fine-tune their software to send less traffic over modems due to the lower available bitrate.
- *Game design:* Different games use different methods to communicate. For example, some games may require *NxN* connectivity in which all participating hosts send messages to all of their peers, while others would consist of a centralized server which receives updates from a number of clients, and then broadcasts the resulting game state. Over a LAN, broadcast or multicast techniques can be used.
- *Game style:* Fast-action "shoot 'em ups" may generate more traffic than slower paced "interactive problem-solving" games. In particular, "shoot 'em up" games require the player to make split second decisions. If the game state is not updated many times per second on the player's screen, the player may not be able to respond (i.e., attack an opponent or dodge a missile) properly. Fast-paced games may require participating hosts to transmit an update periodically every *n* ms, while slower games may only send updates when their state has changed significantly since the last update.
- *User experience and playing style:* Different users play games using different strategies. For example, in "shoot 'em up" games, two main strategies have emerged; some players ("hunters" or "roamers") are continuously on the move, while others ("lurkers" or "campers") tend to wait in one location until they can ambush an opponent. These strategies may result in differing traffic characteristics if the game generates traffic based on state updates rather than periodically.

The value of game traffic models cannot be underestimated. These models could drive background traffic generators in testing environments and be used to determine expected latencies in different hardware

[1] These reasons are not necessarily orthogonal.

environments. With the on-line gaming market soon expected to exceed one billion dollars annually, it is likely that game playing will be a major driving force of Internet evolution.

In this paper we analyze a number of traces of network game traffic. We develop source models of packet size and interarrival periods[2] based on analytical distributions. Our models are descriptive and structural, in the sense that the distribution parameters are often correlated with system characteristics such as CPU speed.

2 Traffic Source Modeling

In this section we provide the motivation for examining source models of game traffic, discuss previous Internet traffic modeling research and introduce our analytical methodology.

2.1 Background

Traditionally, network traffic models have been developed under the assumption that packets or flows arrive according to a memoryless Poisson process [Klei74]. This implies that observed network traffic exhibits only short-term autocorrelations (the autocorrelation function of the observations becomes insignificant after at a finite lag). More recently, *self-similar* or models of network traffic have become dominant in the literature [LTWW94]. These models can capture the long-term autocorrelations (the autocorrelation function of the observations remains significant for all lags) that has been measured in LAN and backbone traffic.

While Poisson models are well understood and can be used to produce tractable analytical results for many complex systems, their underlying assumptions do not always reflect the system being analyzed. Self-similar models are not as well understood, but have been fit to many real-world traffic traces, including VBR video streams [BSTW95] and World-Wide Web traffic [BC97]. However, there is no concise relation that describes how self-similar traffic affects queueing at switches or end-user performance. Additionally, self-similarity implies dependence between observations that are arbitrarily spaced in time. A phenomenological explanation of long-range dependence is not obvious, given the relatively short-term nature of most network transactions (i.e., email and WWW sessions). It has been proposed that self-similarity exists only in an asymptotic sense, and that real network traffic exhibits a finite “correlation horizon,” beyond which dependence is negligible [GB96].

In the search for possible causes of observed self-similarity in network traffic, researchers have found that multiplexing ON/OFF traffic sources with strictly alternating ON and OFF periods will produce self-similar aggregate traffic if either the ON or OFF periods (or both) are drawn from heavy-tailed (infinite variance) distributions [WTSW97]. This type of study is source modeling at the *macro* scale; that is, one or more sessions or flows are modeled as a single ON period. While the existence of infinite-variance distributions has been the subject of some debate, it has been empirically shown that Ethernet hosts [WTSW97] and the upper tail of the distribution of bytes in FTP data transactions [PF95] can be well-modeled with infinite-variance distributions, even though the empirical data analyzed was finite in quantity.

In this paper, we consider source modeling at the *micro* scale; in other words, we examine the interarrival dynamics between individual packets within a gaming session. Micro scale models have been used previously to model the packet interarrival periods of telnet [PF95] and FTP [PJCME92] traffic. In particular, our goal is to fully characterize the intra-session dynamics of network game traffic. In order to do this, we must offer models of packet interarrival periods and packet sizes, as well as the correlations of these statistics. With fully developed source models, we can begin to understand the overall impact of

[2] We use the term "interarrival periods" to refer to the amount of time between sequential packet transmissions by a particular host. These packets can be thought of as "arriving" on the wire, even as they are "departing" from the host.

aggregating (multiplexing onto a single link) the traffic generated by some number of network gaming hosts.

2.2 Methodology

When fitting Internet traffic to analytical models, one quickly discovers that traditional goodness-of-fit tests, such as Chi-square and Kolmogorov-Smirnov, often fail [Paxs96]. It is well known within the statistics community that these tests are biased against large or "messy" data sets, as well as data that exhibit significant autocorrelations [GM83]. The Internet traffic traces that we study are both large (thousands or tens of thousands of samples per trace) and highly variable, many with significant autocorrelation.

Instead of determining whether or not a data set fits some model, we determine the *discrepancy* between the empirical data and the mathematical model. A discrepancy metric measures the magnitude of the departure of the data from the model. In particular we utilize the λ^2 discrepancy measure [PJ90], which has been used in the past [Paxs94] to determine the closeness of the fit between TCP traffic and analytical models. The λ^2 measure, which is described in more detail in Appendix A, returns a non-negative value of discrepancy; in other words, the larger the value, the greater the discrepancy between the data and the model. For a λ^2 value of 0, there is no discrepancy. Using λ^2, we can determine which of several models describe a data set best. We can also compare the relative fit of some data set D1 to model M1 with the fit of some data set D2 to model M2.

However, before we use λ^2, we must decide upon an analytical distribution to fit a data set to. With practice, we have found that we can narrow the range of models to choose from by carefully eyeballing[3] the PDF or CDF of the data. For example, an exponential distribution has a notably different PDF than a Normal or a Pareto distribution. Once a distribution has been chosen, we use a Maximum Likelihood Estimator (MLE) to determine the distribution's parameters[4]. To find the parameter values of a distribution with *n* parameters, we use the first *n* moments of the data set. If we can't easily determine the best analytical distribution to use, we can use the λ^2 measure to find the analytical model from which our data deviates the least.

At this point we can provide visual displays of the quality of the fit by plotting the empirical an analytical PDFs and CDFs on the same graph, or, better yet, comparing the distributions with a quantile-quantile (Q-Q) plot. In a Q-Q plot, an empirical distribution, $F(x)$, is compared to a reference distribution, $G(x)$. For some quantile q_i, we find x_i and y_i such that $F(x_i) = G(y_i) = q_i$, then we plot x_i versus y_i for all i. If the resulting points form a straight line, we can conclude that $F(x) = G(x)$. In practice we often find deviations in the fit. The power of the Q-Q plot is that we can quickly determine *where* those deviations occur (i.e., in the main body, the lower tail, the upper tail, etc.). The Q-Q plot has been used extensively in the networking literature for this purpose (see [Mukh94] and [Paxs97] among others).

When the Q-Q plot indicates a deviation, especially a deviation in the upper or lower tail, we may prefer to model the data set with a *split distribution*. In this case, we model part of the data set with one distribution

[3] Naturally, eyeballing a PDF can be misleading, especially if the PDF is presented in the form of a histogram. However, in many cases the actual values of the distributions that we examined were limited in range. Thus, we were able to use a histogram with as number of bins equal to or greater than the range as a reasonably accurate eyeballing technique for determining the general shape of the distribution.

[4] For some distributions it is possible to determine confidence intervals for the parameters derived from MLE techniques. However, while we found that large confidence intervals generally indicate a poor fit, tight confidence intervals were often reported even when Q-Q plots and λ^2 measures reported a poor fit. Thus, we do not report MLE confidence intervals in this paper, due to their misleading nature.

and the rest of it with another. For example, in [GW94], the marginal distribution of the number of bytes transmitted in a VBR video stream over a fixed period of time is modeled with a Gamma distribution for the lower tail and body, and with a Pareto distribution for the upper tail. Naturally, we can split a distribution as many times as necessary, but more than two or three splits results in a cumbersome analytical model.

The particular nature of the traffic that we analyze often causes "spiky" data sets, in which the empirical distributions are not smooth. Such data is pathologically difficult to fit to any well-known analytical distribution, and will often cause large λ^2 values, even if the analytical distribution fits the general shape of the data set very well. In these cases, we either settle with a large λ^2 value and note the spike(s) that caused it, or split the data set. For the latter, we model the spike(s) as deterministic(s), and indicate the quantiles between which they fall.

Finally, in order for our models to be useful, a group of data sets that were generated by a particular game should all be well-modeled by the same distributions with analytical parameters falling in a relatively limited range. For example, if we model the interarrival periods of a game's traffic with a Gamma distribution, even if the techniques discussed above indicate good fits (i.e., λ^2 is small for all data sets), a wide range of Gamma parameters across different traces will render our model less useful. If we wish to develop traffic generators based on our models, we should present them with limited ranges for model parameters. As shown in Section 3, we were always able to fit a particular game's traffic to a small range of parameter values, and these values were on source host hardware and software characteristics.

2.3 Algorithm

The overall methodology of this study loosely followed this algorithm:

1. Visually examine the PDF or CDF of the data set and choose an appropriate analytical distribution.
2. Use an MLE technique to fit the data set to the distribution.
3. Example a Q-Q plot of the fit. If the fit deviates for a particular portion of the distribution, consider modeling the data set with a split distribution. If the overall fit is poor, start over with a different analytical distribution.
4. Determine the λ^2 value of the fit.
5. Examine extreme upper tail for deviations.
6. Calculate autocorrelation.

2.4 (Un)importance of Outliers and Tail Behavior

In many studies of network traffic, characterizing the outliers (usually in the form of heavy upper tails) of a measured data set may be as important, or even more important, than characterizing the body of the data set. For example, [Paxs94] found that less than 1% of all FTP transactions accounted for over 50% of all traffic transmitted. Likewise, [BSUB98] measured the magnitude of consecutive UDP packet loss and found that less than 1% of all busts of loss accounted for about 33% of all lost packets. In these cases, ignoring the extreme upper tail of a distribution when modeling the observations would lead to a model that does not capture some very important characteristics exhibited in real Internet traffic.

For game traffic, we are concerned with modeling packet interarrivals and size. For both of these statistics, we occasionally found "heavy" upper tails. However, the heaviness of these tails, and the importance of the heaviness, is not as crucial to model. For example, we found several interarrival distributions in which the extreme upper tail was heavy. But the magnitude of this heaviness was minor – the tails consisted of no more than 10 packets, always less than one order of magnitude away from the body of the distribution. In some of these cases, we ignored the extreme tail (< 0.05% of the total distribution) when constructing a model. However, the overall impact and importance of these tails is very minor with respect aggregate traffic dynamics. In Section 3, we explicitly note when tail truncation was performed, the magnitude of the truncation, and when the resulting model underestimates or overestimates the tail weight.

3 Measurements and Analysis

The video game market is highly volatile. New platforms emerge every 2-4 years that render current gaming technology obsolete. Hundreds of games are introduced to the market each year, but few of them enjoy large-scale popularity. A growing fraction of games support multi-player network play, in which two or more people can complete against or cooperate with one another. Usually these games are real-time; the players cannot pause or stop the game without completely exiting it. Different multi-player games offer different connectivity options – some can only be played over modem or null-modem direct connections. Others support IPX and/or TCP/IP protocols. IPX games are played over one or more LANs while TCP/IP games can be played over LANs or the Internet.

Multi-player games can be divided into three general categories: fast-action, slow-action, and strategic. Fast-action games are usually use three-dimensional first-person perspective. They require the player to traverse a virtual reality and make split-second decisions. Popular fast-action games are Quake I & II, Hexen I & II, Jedi Knight and Duke Nukem 3D. Slow-action games are usually 3D but the player's perspective is third person. Each player controls one or a small number of characters. The player has to make split-second decisions on occasion, but spends more time exploring and interacting with other players. Popular slow-action games are NetTrek and Diablo. Strategic games require the player to develop and deploy an army in order to vanquish opponents. Generally, more time is spent planning an attack or defense than on the actual combat. Popular strategic games include Command and Conquer I & II, Total Annihilation, Warcraft, Starcraft, and Age of Empires.

Trace	*Players*	*Clients*	*Duration*	*Date*
Quake-1	4	3	10:26	1/17/98
Quake-2	6	5	9:57	2/4/98

Table 1: Game traffic traces.

Table 1 summarizes the game traces that we collected from Quake I. The measurements were performed using *tcpdump* [JLM] from a Pentium-based PC running FreeBSD 2.2.5. In all cases, *tcpdump* reported that it dropped no packets. The games took place over a set of Ethernet LANs. For our traces, the number of players in the game is one greater than the number of client hosts. This is because one player uses the server host to play the game.

For purposes of understand how game traffic impacts the Internet we ran all games in TCP/IP mode. In order to characterize game traffic on the micro scale, we need to model the first and second order statistics of the interarrival period and the packet size.

3.1 About Quake

Quake is a fast-action game in which a number of players each control a single character. The player traverses a highly graphical maze filled with weapons, ammunition, and opponents. The goal is for each player to kill the other players as many times as possible. When players "die" they are out of the game until they press a key and are "resurrected." Currently, Quake and its successor, Quake II, are the most popular multi-player games. As of the time of this writing, one can find hundreds of Quake servers running on the Internet at any time of night or day.

Quake servers run either in dedicated or non-dedicated mode. The non-dedicated mode requires that one player set up his or her host as the server, and the other players join the game as the clients[5], while the server also acts as client for its player. Dedicated mode is used when a server host is set up just to run the server, and no players use this host as a client.

UDP is used as the transport protocol for all game communication. A client transmit cycle consists of reading a server packet, processing it, rendering the client's current view on the screen, sampling input devices (mouse, keyboard, joystick), then transmitting an update packet (usually 24 bytes) to the server. The most computationally expensive portion of the cycle is the rendering, and, as shown below, will cause slower hosts to transmit far fewer packets to the server. A non-dedicated server will also do server processing (updating the global gaming state) and transmit a variable-length update packet to all clients, back-to-back. Dedicated server transmission is at timer-based intervals, and is configurable to be between 50 and 200 ms. Our study consisted of running servers only in non-dedicated mode.

3.2 Measurements

Host	*CPU / OS / RAM*	*Total Packets*	*mean/stdev of packet size*	*mean/stdev of interarrivals*	*Mean bitrate*
Client 1	266 Mhz PII / Win95 / 64 Mb	38453	23.93 / 1.02 bytes	15.72 / 4.31 ms	11.79 Kbps
Client 2	233 Mhz PII / WinNT / 64 Mb	26324	23.91 / 1.18 bytes	23.39 / 6.61 ms	8.07 Kbps
Client 3	233 Mhz PII / WinNT / 64 Mb	23528	23.89 / 1.18 bytes	26.08 / 6.85 ms	7.22 Kbps
Server	133 Mhz P / WinNT / 64 Mb	36542	109.76 / 42.61 bytes	49.66 / 12.04 ms	51.25 Kbps

Table 2: Description of trace Quake-1.

Host	*CPU / OS / RAM*	*Total Packets*	*mean/stdev of packet size*	*mean/stdev of interarrivals*	*Mean bitrate*
Client 1	133 Mhz P / WinNT / 64 Mb	14431	23.89 / 1.33 bytes	41.37 / 9.86 ms	4.62 Kbps
Client 2	200 Mhz P / Win95 / 32 Mb	32670	23.95 / 0.91 bytes	18.25 / 5.02 ms	10.49 Kbps
Client 3	166 Mhz P / WinNT / 48 Mb	17381	23.91 / 1.17 bytes	34.35 / 8.76 ms	5.57 Kbps
Client 4	266 Mhz PII / Win95 / 64 Mb	36776	23.95 / 0.86 bytes	16.15 / 4.04 ms	11.80 Kbps
Server	233 Mhz PII / WinNT / 64 Mb	21594	77.43 / 29.44 bytes	26.58 / 8.42 ms	22.40 Kbps

Table 3: Description of trace Quake-2.

Table 2 and Table 3 describe the basic observed characteristics of traces Quake-1 and Quake-2, respectively. For each client and server we report the host workstation's processor[6], operating system, and main memory size. "Total packets" is the total number of packets that the host transmitted over the duration of the trace. The first and second moments of the empirical packet size and interarrival

[5] A player may or may not play on the host that is the server. Many Internet game servers are continuously running; thus, all players may be connected with clients.
[6] All processors are of the Intel Pentium family, unless stated otherwise. A "P" is a standard Pentium, and a "PII" is a Pentium II.

distributions are also reported, as well as the overall mean bitrate. For packet size and bitrate calculations, we only considered payload size, not overall packet size including headers.

3.3 Models

We applied the techniques discussed in Section 2 to our Quake traces. We found that for both clients and servers the interarrival periods were well modeled with extreme distributions (see Appendix C). For purposes of illustration, we include (in this section, as well as all others) a graph of some of the client's PDFs and Q-Q plots. Figure 1 shows clients 1-3 of trace Quake-1. The PDF plot histograms are the empirical PDFs and the dash-dot line is the fitted distribution. The Q-Q plot shows the analytical quantiles with a dash-dot line and empirical quantiles each with a "+."

Since Quake servers transmit separate packets to all clients with little or no pause in between, *absolute* server interarrival times would have a mode near 0 ms. Since this behavior is simple to capture in simulations, we instead focus on modeling the time between these transmission "bursts." Thus, the server interarrival data sets have had all interarrival periods of less than 5 ms removed.

From visual analysis of Figure 1, it is clear that both client 2 and client 3 fit their analytical distributions much closer than client 1. This is borne out by the more rigorous λ^2 discrepancy measure: 5.94 for client 1, 0.39 for client 2, and 0.01 for client 3. The huge mode at 13 ms for client 1 could not be captured by its model. As an alternative, we model the lower 50% of client 1 as deterministic at 13 ms, and the upper 50% as exponential with mean at 19.61 ms (see Figure 2). This split distribution allows us to capture the characteristics of the data set much more closely, although the upper 5% of the right tail is overestimated by the model. We applied this technique of splitting distributions to several of our models of interarrival distributions.

We note that client packet size is well-modeled as deterministic (see Table 2 and Table 3); therefore it is not shown graphically. However, server packet size is variable and is fit to an extreme distribution (λ^2 = 0.069) in Figure 3. In Figure 3 we also fit server interarrivals to an extreme distribution (λ^2 =1.261). The relatively poorer fit in the latter case may be due to the large mode at 60 ms that is apparent in Figure 3a.

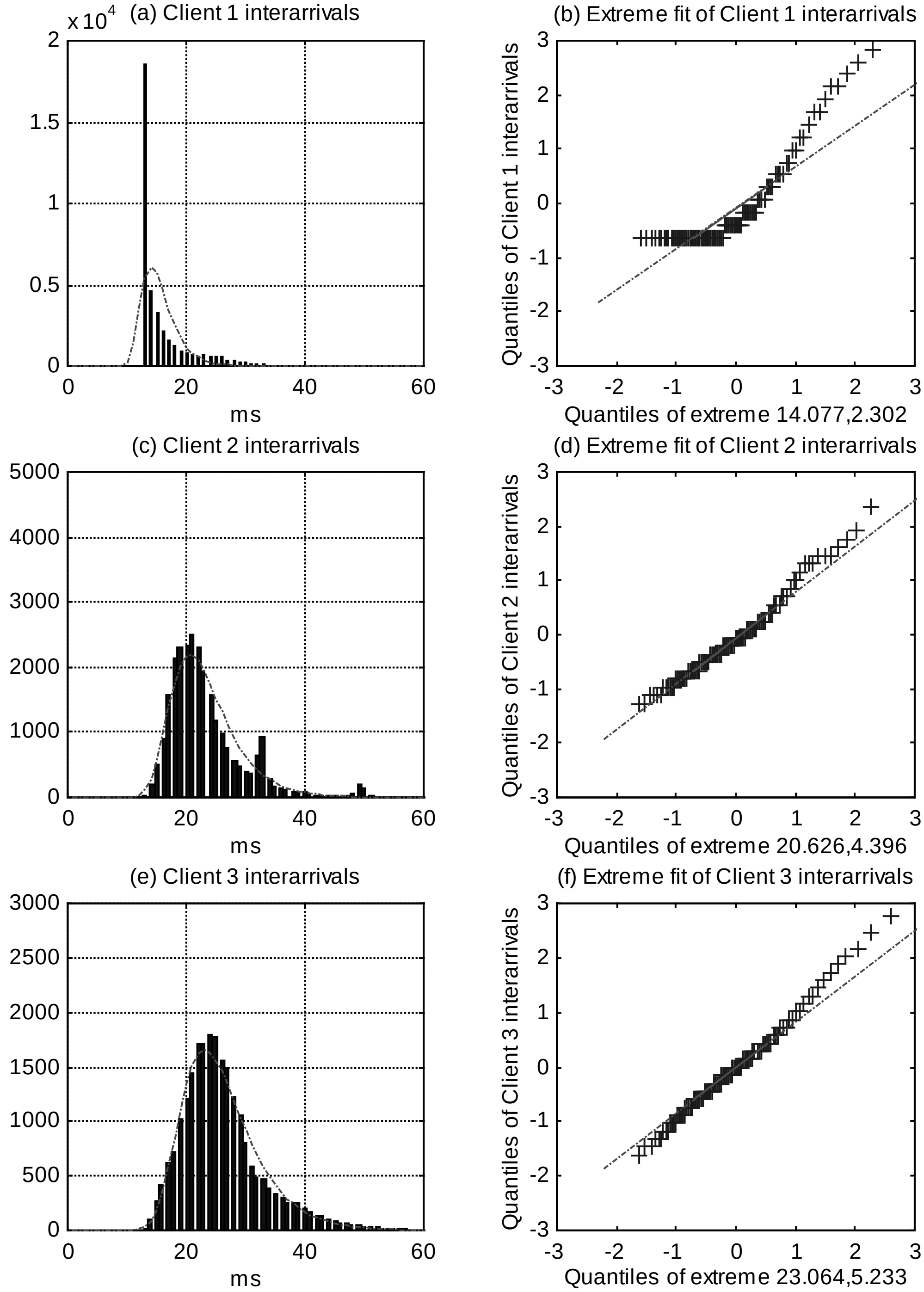

Figure 1: Client interarrival distributions for Quake-1.

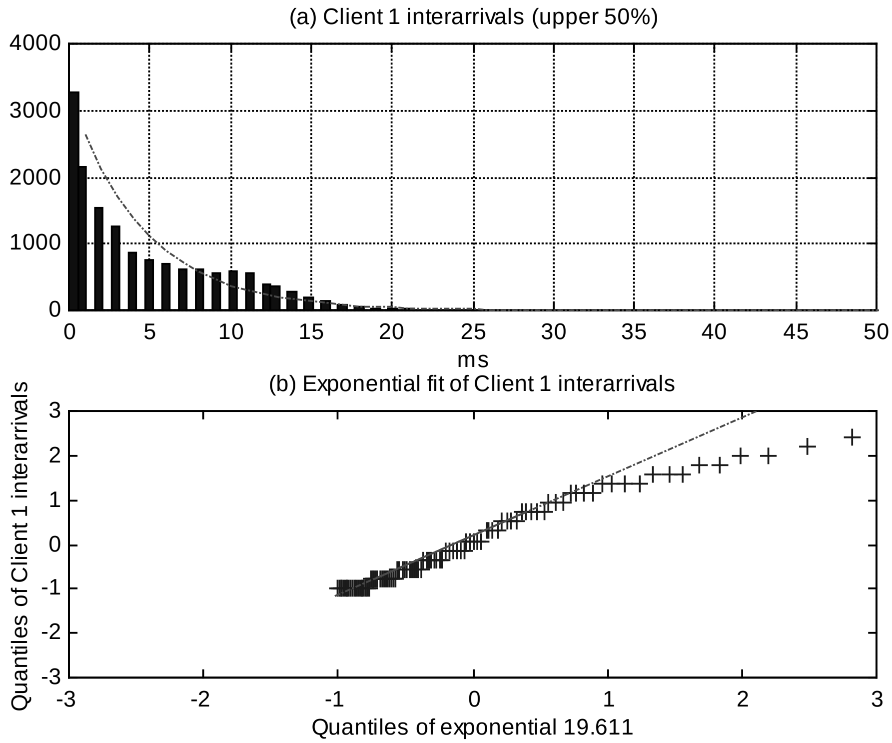

Figure 2: Exponential fit of upper 50% of Client1 interarrivals.

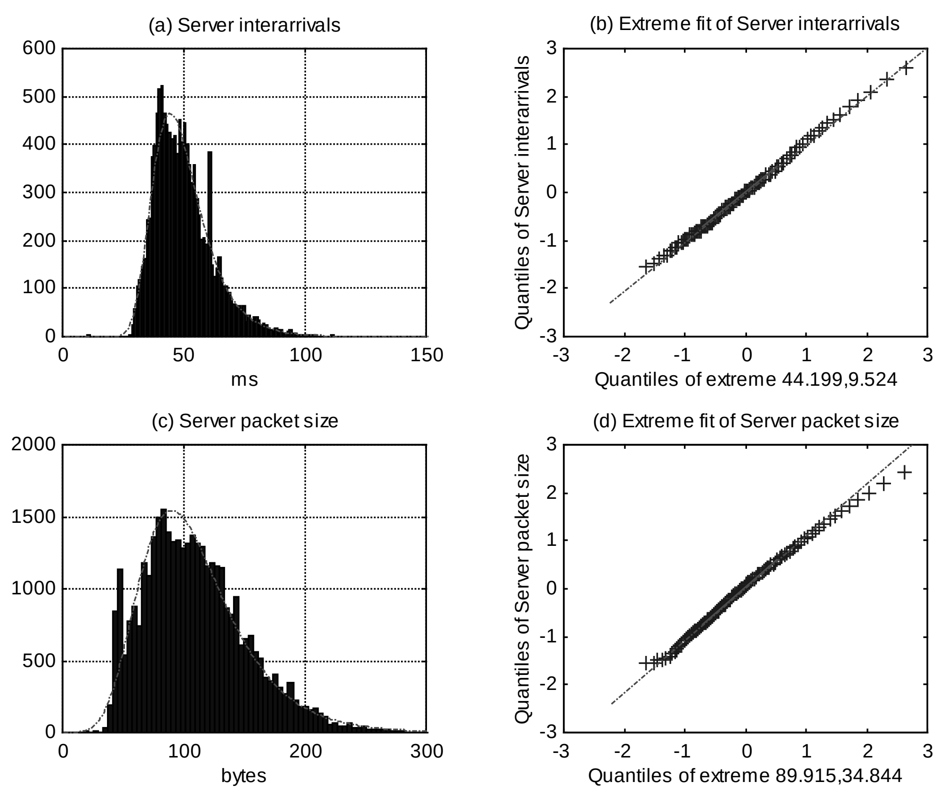

Figure 3: Server interarrival and packet size distributions for Quake-1.

A summary of our models of traces Quake-1 and Quake-2 appear in Table 4 and Table 5. In order to save space, we do not show the distributions and Q-Q plots for the Quake-2 traffic. For each client we show the model used to fit interarrivals and packet size, along with the fitted parameters.

In some cases we found that the upper tail of the interarrival distribution was very heavy. For example, in trace Quake-2, client 3's five greatest interarrival values ranged from 128 ms to 273 ms, even though the overall mean of the distribution was 34.35 ms. Fitting the entire distribution gave us an inflated λ^2 value, due to the large number of empty bins (see Appendix A for a discussion of bin size considerations for λ^2). However, after removing these 5 outliers (only 0.028% of the data), our λ^2 value was more reasonable. We performed this "tail removal" for the interarrivals of several hosts in trace Quake-2. In the "tail" column of Table 4 and Table 5, we use the format *x/y* to denote the number of items removed from the upper tail, and the quality of the tail estimation *after* these items, if any, have been removed. The quality of the tail estimation metric takes on one of three values: "-" if the model underestimated the tail, "+" if the model overestimated the tail, and "0" if the model did a reasonable job of tail estimation. For example, a tail entry of "3/+" indicates that three items were removed from the tail and the resulting model overestimates the tail's weight.

We also are concerned with the serial correlation of interarrivals and packet sizes. Instead of reporting the entire autocorrelation (ACF) function for each host, we show the autocorrelation at lag 1 only. For all hosts we found that while packet sizes were not significantly correlated, interarrivals were. The greatest autocorrelations per trace were produced by the hosts with the fastest and slowest CPUs (client 1 in trace Quake-1, client 1 and client 4 in trace Quake-2). For server autocorrelation, we found that for both interarrivals and packet size, there was a strong positive correlation on a per-client basis. The "ACF(1)" column has the format *x'/y'* where *x'* is the autocorrelation at lag 1 for the entire trace and *y'* is the autocorrelation on a per client basis. The fact that per-client packet sizes are more strongly correlated than the trace's raw packet sizes indicates that a different amount of information is being transmitted to each client.

Data	*Model*	*Parameters*	λ^2	*Tail*	*ACF(1)*
Client 1 interarrivals	Deterministic (lower 50%)	a = 13	0.247	0/0	0.95
	Exponential (upper 50%)	a = 19.61	0.292	0/-	
Client 1 packet size	Deterministic	a = 24	0.004	0/0	0.00
Client 2 interarrivals	Extreme	a = 20.63, b = 4.39	0.396	0/-	0.87
Client 2 packet size	Deterministic	a = 24	0.005	0/0	0.00
Client 3 interarrivals	Extreme	a = 23.06, b = 5.23	0.017	0/-	0.87
Client 3 packet size	Deterministic	a = 24	0.006	0/0	0.00
Server interarrivals	Extreme	a = 44.20, b = 9.52	1.261	0/0	-0.45/0.94
Server packet size	Extreme	a = 89.92, b = 34.84	0.069	0/+	0.25/0.85

Table 4: Models for trace Quake-1.

Data	*Model*	*Parameters*	λ^2	*Tail*	*ACF(1)*
Client 1 interarrivals	Extreme	a = 36.38, b = 8.01	0.768	2/-	0.84
Client 1 packet size	Deterministic	a = 24	0.007	0/0	0.00
Client 2 interarrivals	Extreme	a = 15.89, b = 2.98	0.000	0/0	0.62
Client 2 packet size	Deterministic	a = 24	0.003	0/0	0.00
Client 3 interarrivals	Extreme	a = 30.26, b = 6.30	0.458	5/-	0.60
Client 3 packet size	Deterministic	a = 24	0.005	0/0	0.00
Client 4 interarrivals	Deterministic (lower 50%)	a = 14	0.328	0/0	0.78
	Exponential (upper 50%)	a = 18.38	0.000	0/-	
Client 4 packet size	Deterministic	a = 24	0.003	0/0	0.00
Server interarrivals	Extreme	a = 22.84, b = 5.29	0.357	0/-	-0.59/0.58
Server packet size	Extreme	a = 65.58, b = 19.12	0.820	10/-	0.08/0.78

Table 5: Models for trace Quake-2.

4 Discussion

Using either extreme or split deterministic/exponential distributions, we have captured the most relevant characteristics of Quake client and server interarrivals as well as server packet size. Client packet size is well-modeled as deterministic. For most of the non-deterministic distributions, our models slightly underestimated the weight of the extreme upper tail.

From the tables in Section 3.3, we find that the parameters of the client interarrival distributions are highly correlated with the CPU speed of the hosts. From these results, it is clear that the client rendering speed has a large impact on transmission rate. The slower hosts have significantly higher and more variable interarrivals, while the fastest hosts transmit most packets at 13 or 14 ms intervals. This host-wise variability challenges the structure of mathematical analysis of LAN traffic which assume that all hosts exhibit statistically identical source characteristics.

Given our models, the next step is to develop synthetic traffic generators that can be used to produce parameterized game traffic for simulation purposes. Such generators will be used to test remote access servers (modem pools) as well as LAN switches. From the results of the previous section, the empirical distributions and parameters to use are clear (see Appendix C for a discussion of generating random numbers with an extreme distribution). However, generating any random deviate with a given ACF or first lag autocorrelation is an open problem. A promising approach that we are currently exploring involves use of TES processes [Mela93].

5 Conclusions and Future Work

The most significant contribution of this study is a characterization of observed network traffic generated by a popular fast-action game. We have also proposed a source model that can be used to generate game traffic for background testing loads. This model, based on extreme and split deterministic/exponential distributions, should be explored as an alternative to the deterministic or exponential interarrival distributions that are commonly used in queueing analysis.

On the other hand, the scope of this study was very small – only one measurement of a single game was analyzed. While Quake is extremely popular, it may not be representative of network games in general. The results of this study should therefore be considered to apply only to Quake, though it is likely that games with a similar theme and style, such as Doom or Quake II, may exhibit similar traffic patterns. Preliminary studies of Quake II and NetTrek traffic have resulted in differing models, though our general characterization framework was very successful in modeling these games as well.

We plan on continuing and enhancing this study in the following ways:

- More traffic observations of Quake. In particular, we would like to determine how well we have characterized the source models of the client and server. We would also like explore how the number of clients per game session influenced traffic patterns.
- Traffic observations of other network games, using both highly-interactive shoot 'em ups like Quake, as well as less interactive games.
- Development of tools to synthetically generate game-like traffic.

6 Appendix A: The λ^2 Discrepancy Measure

The λ^2 discrepancy measure was introduced in [PJ90]. It facilities the determination of how close a data set fits an analytical model. Applying λ^2 to network data has been discussed in detail in [Paxs94], and the reader is referred to that work for a complete discussion of the measure and relevant application issues. Here, we summarize how we used λ^2 and our observations of its utility in evaluating game traffic models. The λ^2 metric is defined as

$$\lambda^2 = \frac{X^2 - K - df}{n-1}$$

where *n* is the number of samples in the data set and *df* is the number of degrees of freedom of the test. For us, this is the number of X^2 bins minus one, minus the number of parameters used estimate the analytical distribution. Let E_i be the number of items in bin *i* of the empirical data set and T_i is the number of items in bin *i* of the analytical distribution. *K* is given by

$$K = \sum_i \frac{E_i - T_i}{T_i}$$

Since it is based on the X^2 goodness-of-fit statistic, λ^2 requires the use of X^2. This in turn requires a method for binning a continuous distribution. A number of methods are available (some are discussed in [PJ90]). As in [Paxs94], we used the technique introduced in [Scot79], which chooses an optimal number of fixed-size bins, with a width, *w*, given by

$$w = 3.49\sigma n^{-1/3}$$

where σ is the empirical standard deviation and *n* is the number of observations in the data set. The motivation for using fixed-width bins, as opposed to equiprobable bins[7], is that, in the latter case, a long-tailed distribution will often have its entire tail lumped into a single bin. This would give us a overly-optimistic metric for how well we fit the tails of the empirical data.

Our experience with λ^2 has largely confirmed the observations of [Paxs94]. The λ^2 measure is unreliable in when the number of bins is large. In particular, fitting heavy upper tailed distributions tends to produce very large values of λ^2. [Paxs94] suggested that in these circumstances, the data be log-transformed so that the tail (and the number of bins) is reduced. This technique did not always work for our data – sometimes we could not find a reasonable model of the log-transformed version. Given that both the magnitude and importance of upper tails is far less in our data than that studied in [Paxs94], we chose to truncate the upper tails of our distributions when they were preventing a good fit. For example, by

[7] Equiprobable binning is discussed in depth in [DS86].

removing the greatest 2 – 10 values[8] of some of the client interarrival distributions in the Quake-2 trace, we found that the optimal number of bins was reduced dramatically and λ^2 was brought from values over 1000 (!) to under 1. Naturally, we noted when tail truncation was necessary (see Section 3).

When comparing λ^2 to the visual fits given by Q-Q plots, we discovered that when the empirical data did not have long tails, λ^2 was not very sensitive to deviations in the tails. This can be seen in the fit of client 3's interarrivals to an extreme distribution in Section 3.3 – the data set had a lighter lower tail and a heavier upper tail than the analytical model, but λ^2 reported a discrepancy of only 0.01, which is very small.

When binning data, it is always important to understand the impact of bin size on the goodness-of-fit or discrepancy metric. It is well-known that X^2 is very sensitive to bin choice. We found that λ^2 is relatively insensitive (λ^2 might fluctuate by up to a factor of 2 or 3, but not much more) to bin size if it is close to the optimal value described above. This would indicate that the [Scot79] technique for determining an optimal bin size is reasonably efficient. Naturally, smaller data sets were more sensitive to bin size.

7 Appendix B: Using X^2 and λ^2 with Deterministic Distributions

The chi-square (X^2) goodness of fit test is defined as follows. Given an empirical distribution (E) and a theoretical distribution (T), divide each of these distributions into N bins. Let E_i be the number of observations in bin i for the empirical distribution and T_i be the *expected* number of observations in bin i for the theoretical distribution. X^2 is defined as

$$X^2 = \sum_{i=1}^{N} \frac{(E_i - T_i)^2}{T_i}$$

For deterministic distributions, in which all observations are expected to have the same value, this equation causes a divide-by-zero ambiguity if the empirical data set contains values that vary from the expected value where the expected value is zero. To rectify this problem, we use the following alternative version of chi-square

$$\hat{X}^2 = \sum_i \frac{(T_i - E_i)^2}{E_i}$$

Evaluated only where the empirical data set is non-zero, this metric performs equivalently to standard X^2 where T_i is non-zero and, is E_i where $T_i = 0$. In order to compute a λ^2 (see Appendix A) based on this metric, we also modify K such that

$$\hat{K} = \sum_i \frac{T_i - E_i}{E_i}$$

[8] These numbers were always less than 0.05% of the total number of samples.

Thus, $\hat{\lambda}^2$ can be computed the same as λ^2, but using $\hat{X}^2$ and $\hat{K}$ in place of X^2 and K, respectively. In practice we have found that $\hat{\lambda}^2$ provides intuitively pleasing results – it grows in proportion to the deviation of the empirical data set. It may produce results that are more optimistic than the X^2 technique usually produces, but for comparing the goodness-of-fit between different data sets and distributions, it suits our purposes.

8 Appendix C: The Extreme Distribution

The extreme distribution is well-suited to modeling data sets with significant modes. The extreme CDF is given by [DS86]

$$F(x) = \exp\left[-\exp\left[-\left(\frac{x-a}{b}\right)\right]\right]$$

Taking the first derivative of the CDF, we get the extreme PDF

$$f(x) = \frac{dF(x)}{dx} = \left(\frac{1}{b}\right)\exp\left[-\exp\left[-\left(\frac{x-a}{b}\right)\right]\right]\exp\left[-\left(\frac{x-a}{b}\right)\right]$$

Parameters *a* and *b* are roughly correlated with the mode and the variability of the distribution, respectively. The maximum likelihood estimator for fitting a data set *X* with *n* samples to the extreme distribution is given by [JK70]

$$b = \frac{1}{n}\sum_{i=1}^{n} X_i - \left[\sum_{i=1}^{n} X_i \exp\left(-X_i/b\right)\right] \Big/ \left[\sum_{i=1}^{n} \exp\left(-X_i/b\right)\right]$$

$$a = -b\log\left[\left(\frac{1}{n}\right)\sum_{i=1}^{n}\exp\left(-X_i/b\right)\right]$$

Given the above, we iterate to solve for *b* and then use the result to find *a*.

For simulation purposes, we must be able to generate random variable with an extreme distribution. This can be accomplished by solving the CDF for *x* and replacing *F(x)* with *U(0,1)*

$$x = a - b\ln\left[-\ln\left(U(0,1)\right)\right]$$

where *U(0,1)* indicates random variable drawn from a continuous uniform distribution bounded by 0 and 1.

A distribution is said to be *log-extreme* if, after applying a logarithmic transform to the data, it exhibits an extreme distribution. Log-extreme distributions have been used to model the number of originator bytes in a telnet session [Paxs94].

9 Acknowledgements

This research would not have been possible without the assistance of a number of individuals. We thank Mike Andre, Charles Rygula, Jim Curran and the other members of the 3Com Chicago-based Technical

Communications division who graciously facilitated our traffic measurements. Guido Schuster assisted debugging our Matlab scripts and developing the $\hat{\lambda}^2$ metric. We'd also like to thank John Cash from ID Software for his comments and corrections regarding the operation of Quake clients and servers.

10 References

1. [BC97] M. E. Crovella and A. Bestavros, "Self-Similarity in the World-Wide Web: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, pp. 835-846, Dec. 1997.
2. [BSTW95] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic," *IEEE Transactions on Communications*, Vol. 43, pp. 1566-1579, 1995.
3. [BSUB98] M. S. Borella, D. Swider, S. Uludag, G. Brewster, "Internet Packet Loss: Measurement and Implications for End-to-End QoS," *Proceedings, International Conference on Parallel Processing*, Aug. 1998.
4. [DS86] R. B. D'Agostino and M. A. Stephens, *Goodness-of-Fit Techniques*, Marcel-Dekker, Inc., 1986.
5. [G114] International Telecommunications Union, *Recommendation G.114: One-Way Transmission Time*, Jun. 1996.
6. [GB96] M. Grossglauser and J-C. Bolot, "On the Relevance of Long-Range Dependence in Network Traffic," *Proceedings, ACM SIGCOMM '96*, pp. 15-24, Sep. 1996.
7. [GM83] L. J. Glesser and D. S. Moore, "The Effect of Dependence on Chi-Squared and Empiric Distribution Tests of Fit," *Annals of Statistics*, Vol. 11, pp. 1100-1108, 1983.
8. [GW94] M. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," *Proceedings, SIGCOMM '94*, pp. 269-280, Sep. 1994.
9. [Jain91] R. Jain, *The Art of Computer Systems Performance Analysis*, Wiley, 1991.
10. [JLM] V. Jacobson, C. Leres, and S. McCanne, *tcpdump*, ftp://ftp.ee.lbl.gov/tcpdump.tar.Z
11. [JK70] N. L. Johnson and S. Kotz, *Distributions in Statistics, Volume 1: Continuous Univariate Distributions*, Houghton-Mifflin, 1970.
12. [Klei74] L. Kleinrock, *Queueing Systems Vol. 1: Theory*, Wiley and Sons, 1974.
13. [LC] Lagcity WWW page, http://lagcity.stomped.com/.
14. [LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (extended version)," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, pp. 1-15, Jan. 1994.
15. [Mela93] B. Melamed, "An Overview of TES Processes and Modeling Methodology," in *Performance Evaluation of Computer and Communications Systems*, (L. Donatiello and R. Nelson, Eds.), 359-393, Lecture Notes in Computer Science, Springer-Verlag, 1993.
16. [Mukh94] A. Mukherjee, "On the Dynamics and Significance of Low-Frequency Components of Internet Load," *Internetworking: Research and Experience*, Vol. 5, pp. 163-204, 1994.
17. [Paxs94] V. Paxson, "Empirically-Derived Analytical Models of Wide-Area TCP Connections," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 4, pp. 316-336, Aug. 1994.
18. [Paxs96] V. Paxson, "An Introduction to Internet Measurement and Modeling," *tutorial given at ACM SIGCOMM '96*.
19. [Paxs97] V. Paxson, "Measurements and Analysis of End-To-End Internet Dynamics," Ph.D. Thesis, UC Berkeley, Jun. 1997.
20. [PF95] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, pp. 226-244, Jun. 1995.
21. [PJ90] S. P. Pederson and M. E. Johnson, "Estimating Model Discrepancy," *Technometrics*, Vol. 32, No. 3, pp. 305-314, Aug. 1990.
22. [PJCME92] P. Danzig, S. Jamin, R. Caceras, D. Mitzel and D. Estrin, "An Empirical Workload Model for Driving Wide-Area TCP/IP Network Simulations," *Internetworking: Research and Experience*, Vol. 3, No. 1, pp. 1-26, Mar. 1992.
23. [Scot79] D. W. Scott, "On Optimal and Data-Based Histograms," *Biometrika*, Vol. 66, pp. 605-610, 1979.

24. [WTSW97] W. Willinger, M. S. Taqqu, R. Sherman, and D. Wilson, "Self-Similarity Through High Variability: Statistical Analysis of Ethernet Traffic at the Source Level," *IEEE/ACM Transactions on Networking*, Vol. 5, No. 1, pp. 71-86, Feb. 1997.