Boston University

OpenBU **http://open.bu.edu**

Computer Science CAS: Computer Science: Technical Reports

2002-05-10

A Hierarchical Characterization of a Live Streaming Media Workload

https://hdl.handle.net/2144/1660

Downloaded from DSpace Repository, DSpace Institution's institutional repository

A Hierarchical Characterization of a Live Streaming Media Workload*

Eveline Veloso Virgílio Almeida Wagner Meira
eveline@dcc.ufmg.br virgilio@dcc.ufmg.br meira@dcc.ufmg.br
Computer Science Department
Federal University of Minas Gerais, Brazil

Azer Bestavros Shudong Jin
bestavros@cs.bu.edu jins@cs.bu.edu
Computer Science Department
Boston University, MA, USA

BUCS-TR-2002-014
May 10, 2002

Abstract

We present what we believe to be the first thorough characterization of *live* streaming media content delivered over the Internet. Our characterization of over five million requests spanning a 28-day period is done at three increasingly granular levels, corresponding to clients, sessions, and transfers. Our findings support two important conclusions. First, we show that the nature of interactions between users and objects is fundamentally different for live versus stored objects. Access to stored objects is *user driven*, whereas access to live objects is *object driven*. This reversal of active/passive roles of users and objects leads to interesting dualities. For instance, our analysis underscores a Zipf-like profile for user interest in a given object, which is to be contrasted to the classic Zipf-like popularity of objects for a given user. Also, our analysis reveals that transfer lengths are highly variable and that this variability is due to the stickiness of clients to a particular live object, as opposed to structural (size) properties of objects. Second, based on observations we make, we conjecture that the particular characteristics of live media access workloads are likely to be highly dependent on the nature of the live content being accessed. In our study, this dependence is clear from the strong temporal correlations we observed in the traces, which we attribute to the synchronizing impact of live content on access characteristics. Based on our analyses, we present a model for live media workload generation that incorporates many of our findings, and which we implement in GISMO [19].

Keywords: Live streaming content delivery; streaming media characterization; synthetic workload generation.

1. Introduction

Motivation:. The use of the Internet as a channel for the delivery of streaming (audio/video) media is paramount. This makes the characterization and synthetic generation of streaming access workloads of fundamental importance in the evaluation of Internet and streaming delivery systems.

Over the last few years, there have been a small number of studies that attempted to characterize streaming media workloads [1, 2, 3, 11, 21, 26]. However, to our knowledge, all these studies targeted pre-recorded, stored streaming media objects (e.g., news clips, film trailers, educational clips) and none has considered the characterization of *live* streaming media (e.g., camera feeds). This paper provides such a characterization for a unique data set capturing hundreds of thousands of live streaming media sessions served over the Internet to thousands of users as a complement to a very popular "reality TV show" in Brazil.

While an interesting subject on its own, the characterization of live streams on the Internet is likely to be of paramount importance given the increasing role of the Internet as a delivery channel for live content that *complements* other broadcast channels (e.g., TV). By complementing other broadcast channels, we mean that the Internet enables users to bypass the editing (or "montage") necessary for broadcast purposes (e.g., enabling a user to fix the source of a feed to a specific camera—say goalkeeper view in a soccer game). Enabling this level of access in a scalable manner is a capability that is unique to the Internet architecture (as opposed to broadcast media).[1]

While workload characterization is an important ingredient of performance evaluation and prediction in general, it is particularly critical for proper capacity planning of live (as opposed to stored) content delivery infrastructures (e.g., servers, network, CDN, etc.) To elaborate on this point, note that when dealing with stored content, if the aggregate load on an underprovisioned resource—say a server—reaches a given limit, the server may opt to simply "reject" new requests. This "admission control" solution may be acceptable since a user can be expected to come back at a later time to request the stored content. For live content, turning down a user's request amounts to denying access, since the value of the content is in its liveness. Thus, admission control is not a viable alternative for content providers (or their proxies, such as CDNs) when dealing with enabling their paying customers[2] access to live streaming media content. Capacity planning based on accurate understanding of workload characteristics [22] becomes a necessity. A case in point

*This work was partially supported by NSF research grants ANI-9986397 and ANI-0095988.

[1] Indeed, this lack of editorial controls is the *raison d'être* of the Internet which has catalyzed its growth as a complement to traditional brokers of information exchange (e.g., TV, publishers, news agencies, etc.)

[2] Note that many content providers are now charging for access to streaming content—e.g., CNN's NewsPass [12] and Real Networks' RealOne SuperPass [25] subscription services.

is the experience of thousands of users in January 1999 when attempting to view VictoriaSecret.com's highly-advertised webcast.

Characteristics of Live versus Stored Streams:. The characteristics of live streaming workloads are likely to be fundamentally different from those of pre-recorded, stored clips. For starters, live streaming workloads are likely to exhibit stronger temporal (e.g., diurnal) patterns that may not be present (or may be significantly weaker) otherwise. Also, the range of operations possible with stored media (e.g. VCR functions) are simply not available for live media. More importantly, the correlations between various variables may be significantly different for live and stored media. For example, consider the possible correlation between the length of time a user may be viewing a stream and the QoS of the playout resulting from available network bandwidth. For stored media, one would expect a positive correlation between these two characteristic properties of the workload; namely, users tend to stop viewing a stream when QoS degrades below a certain threshold. For live streams, this correlation may be much weaker and/or the mitigating QoS threshold may be significantly different since users do not have the option of revisiting the content again in the future (as is the case with stored media).

The above-mentioned differences between live media and stored media access patterns stem from the fundamentally different passive versus active roles that users and objects play in each case. Accesses to pre-recorded, stored media objects are *user driven*; they are directly influenced by user preferences—namely, *what* to access and *when* to do so. Accesses to live media are *object driven*; they are directly influenced by aspects related to the nature of the object—e.g., show time, activities captured by various feeds, etc. In such an environment, users are mostly "passive"; they are fairly limited in how they are allowed to interact with objects. Namely, they can only join or leave the audience of the live "active" object.

Paper Overview:. The remainder of this paper is organized as follows. In Section 2, we describe the source of the logs considered in this paper. We present basic information and statistics related to the traces we collected and we introduce the terminology we adopt for the remainder of the paper. In the following three sections, we present results of our characterization along three increasingly granular levels of abstractions, corresponding to client behavior and arrival processes (in § 3), session characteristics (in § 4) and object request characteristics (in § 5). While at this time we are unable to release to the research community the proprietary logs we used in our study, we have parametrized GISMO [19]—a streaming workload generator—to allow the synthetic generation of live streaming content workloads that resemble those we characterize in this paper. This is described in Section 6. In Section 7, we present an overview of related work. We conclude in Section 8 with a summary of our findings and with directions for future work.

2. Live Streaming Workload

2.1 Source of the Workload

We obtained logs of over one-month-worth of accesses to a very popular live streaming media server operated by one of the top ten content service providers in Brazil. This server (a Microsoft Media Server [13]) enabled users to tap into one or both of two live streaming media objects associated with a popular Brazilian "reality TV show" that aired in early 2002 and lasted for 90 days. At any point in time, each one of these live streams provided (audio and video) feeds captured from one of 48 different cameras embedded in the environment surrounding the contestants in the reality show.

2.2 Characterization Hierarchy and Terminology

Requests for live streaming media are presented to the streaming servers in an interleaved fashion. In order to understand the characteristics of this type of workload as well as the hidden structures existing in the interaction between users and live streaming media services, we adopt a hierarchical approach to the characterization of the workload [23]. To that end, we look at the live streaming media workload as a hierarchy of layers. At the lowest layer, the streaming servers receive requests from multiple clients. At the next level up, requests from individual clients can be grouped into sessions. At the top level, sessions from individual clients can be grouped into a client behaviour level.

Throughout this paper, we use the term *live objects* or simply *objects* to refer to live streams (i.e., "continuous" feeds) whose existence is defined by the duration of an event (e.g., live show or game). We characterize access to such objects at three increasingly granular levels of abstractions (or layers), corresponding to *clients*, *sessions*, and *individual transfers*. Within each layer, an analysis of statistical and distributional properties of variables within that layer is conducted. Our approach is to analyze each layer individually in order to obtain a characterization of the arrival processes meaningful for that layer (e.g., interarrival times, level of concurrency), access patterns in that layer (e.g., ON/OFF times), and other statistics (e.g., popularity and temporal correlations).

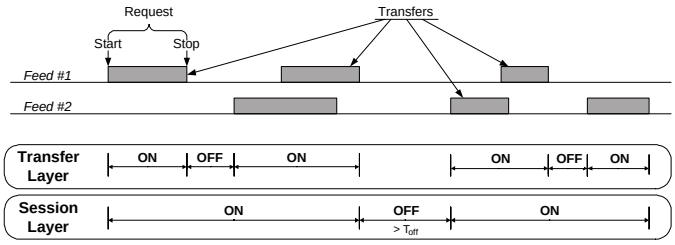

Figure 1: Relationship between client activities and ON/OFF times at the session and transfer layers

Client Layer:. The top layer of our hierarchy focuses on the characteristics of the client population. We identify a client by the unique *player ID* field that is recorded as part of every entry in the logs. Notice that a client corresponds loosely to an individual user. Exceptions to this include cases in which the same software client is used by multiple users sharing the same client machine. Client characteristics we consider include the number of clients accessing the live content (i.e., level of concurrency) over time, client interarrival times, and the relationship between a client's "interest" in the live content (relative to all other clients) and the frequency of access by that client, measured in total number of sessions of (or transfers to) that client.

Session Layer:. Focusing on an individual client, we move to the second layer of our hierarchy, in which we characterize the variables governing client sessions of activity. We define a client session as the interval of time during which the client is actively engaged in requesting (and receiving) live objects that are part of the

same service (e.g., part of the same show) such that the duration of any period of no transfers between the server and the client does not exceed a preset threshold T_{off}. According to this definition, a given client's access pattern is governed by periods of activity (session ON time) and of inactivity (session OFF time). Figure 1 shows how client activities (namely start/stop requests for specific objects) result in various session ON and OFF times.

Transfer Layer:. Zooming in on session ON times, we characterize the bottom layer of our hierarchy, which focuses on individual unicast data transfers, each of which is the result of specific actions performed by a client. Specifically, for live objects, a transfer is the result of a pair of requests to "start" and eventually "stop" viewing a live object.[3] Thus, a given session is characterized by periods of data transfer (transfer ON time) and of silence (transfer OFF time). During transfer ON times, a client is served one or more live objects (e.g., different live views). During transfer OFF times (which by definition must be smaller than T_{off}) no live objects are served to the client. Transfer OFF times correspond loosely to "think" times or to what has been termed "active OFF" times in [15]. Figure 1 shows how client activities result in various transfer ON and OFF times. In this layer, and in addition to characterizing transfer ON and OFF times, we also characterize individual transfer lengths, number of concurrent transfers across all clients, transfer interarrival times, as well as the temporal correlation of transfer arrivals.

Characterizing the workload at these distinct levels of abstraction allows one to concentrate on the analysis of the behavior of the different players that interact in this type of environment—namely *clients* and *objects*. This hierarchical characterization can also be used to capture changes in client behavior and map the effects of these changes to the lower layers of the hierarchical model—i.e., session and transfer layers. Finally, this layered approach enables us to develop an explicable process via which we can generate synthetic live streaming workloads (as we discuss in Section 6).

2.3 Basic Log Statistics and Server Configuration

Table 1 summarizes the basic information and statistics about the logs we analyze in this paper.

Log period	28 days in early 2002
Total # of live objects	2
Total # of client ASs	$1,010$
Total # of client IPs	$364,184$
Total # of users	$691,889$
Total # of sessions	$> 1,500,000$
Total # of transfers	$> 5,500,000$
Total content served	> 8 TeraBytes

Table 1: Basic statistics of the trace used in this paper

The Windows Media Server was configured to enable full logging of all user activities throughout the 28 days of the log collection period. Logs were harvested daily (at midnight). Each entry in the log identifies a single client/server request/response. While the Windows Media Server supports both unicast and multicast services, only unicast transfers were enabled. For each entry in the log, the following information is provided:[4]

1. Client identification—e.g., *IP address, player ID*,
2. Client environment specification—e.g., *OS version, CPU*,
3. Requested object identification—e.g., *URI of requested stream*,
4. Transfer statistics—e.g., *packet loss rate, average bandwidth*,
5. Server load statistics—e.g., *server CPU utilization*,
6. Other information—e.g., *referer URI, HTTP status*, and
7. Timestamp in seconds of when log entry was generated.

Given the coarse one-second resolution of timing information in the server log, it is often the case that *zero* time intervals would be measured—e.g., for ON/OFF times, interarrivals, etc. Throughout the paper, to enable the display of such measurements on a logarithmic scale, we have opted to use the function $\lfloor t+1 \rfloor$ to represent a time measurement of t seconds.

2.4 Log Sanitization

We have identified a number of problems with a small percentage of the entries in the logs we used. Specifically, a number of entries identified request/response activities that span durations longer than the 28-day period of the trace! We suspect that these entries correspond to accesses that spanned multiple log harvests. These requests were excluded from our characterization.

As will be evident later in the paper, there are periods of time during which the number of users accessing content from the server is very large (e.g., few thousands). Thus an important question relates to whether the characteristics we present are influenced by the system's overall capacity. For example, given the feedback nature of the interaction between a user and the system, an overloaded server may "slow down" user activities, or even turn away users, and thus impact our characterization of (say) user interarrivals or the level of concurrency, etc. To ensure that the characteristics we present throughout the paper are not affected by server overloads, we have analyzed the logs and indeed established that periods of server overloads are extremely rare. Specifically, we took all CPU load measurements, as reported in the server logs, and averaged them in one-second bins. The results indicated that the server utilization was below 10% for over 99.99% of the time. Similarily, the server load was below 10% for over 99% of all transfers in the log.

3. Client Layer Characteristics

In this section we present various client characteristics, including number of clients over time (or level of concurrency), the relationship between frequency of access and a client's relative "interest" in the live streaming service, as well as other statistics related to the client population in general.

3.1 Client Topological and Geographical Distribution

An important question that is often asked regarding workload characterization studies has to do with the "representativeness" of the workload. As evident from Table 1, the workload we characterize in this paper is fairly large in terms of the number of clients (as identified by the ID of the software player on the user machine) and the number of accesses made by these clients. Using the IP address of a client in a given session, we are able to map the client population to over 1,000 different Internet Autonomous Systems (AS'es) scattered over 11 countries. Figure 2 shows the "popularity" of each AS in our workload as measured by the number of transfers (left) and IP addresses it commanded (center) that have been traced back

[3] For stored video, other requests may include VCR functionalities (e.g., "pause", "fast-forward", "rewind", etc.)

[4] For details consult the Windows Media Services documents [13].

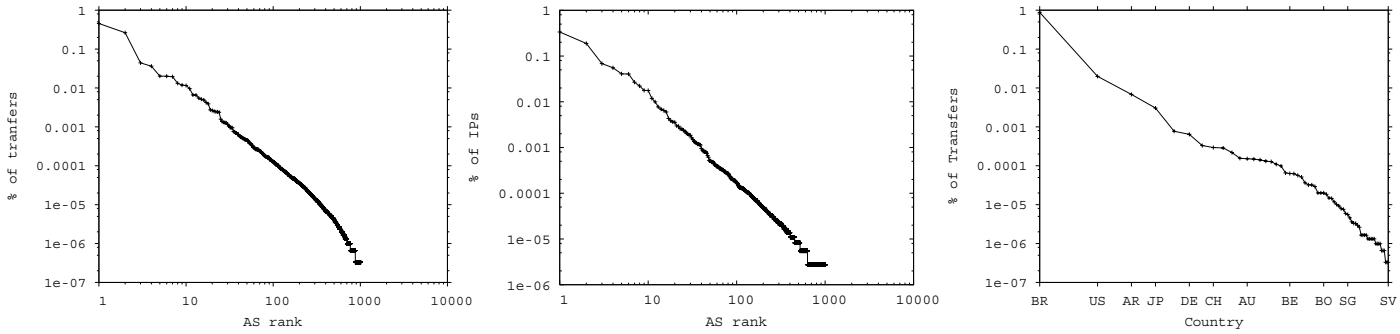

Figure 2: Client diversity: Transfers over AS'es (left), IP addresses over AS'es (center), and transfers over countries (right)

to that AS.[5] Figure 2 (right) shows the distribution of transfers over the various countries.

3.2 Client Concurrency Profile

At any point in time t, there are a number of clients $c(t)$ that are considered active, in the sense that their sessions are still on-going. This level of concurrency could be used to gauge the popularity of the particular content being transmitted at time t. Figure 3 shows the marginal distribution of $c(t)$ over the entire duration of the trace.

Notice that many factors may contribute to the wide variability observed in the number of concurrently active clients, including specific activities occuring within the reality show, as well as diurnal effects on the live content (e.g., no interesting contestant activities between 4am and 11am) and on the client population (e.g., users flock to the site in early evening hours or on weekends). Figure 4 (left) shows the average value of $c(t)$ calculated for consecutive 900-second bins, over the entire period of the trace. Also, in Figure 4, we show the periodic behavior of $c(t)$ by plotting $c(t \mod p)$, where p is one week (center) and one day (right). While the number of clients in the system varies with respect to the day of the week (e.g., weekends have slightly higher average number of clients than weekdays), Figure 4 (right) indicates that diurnal patterns seem to be the main source of variability, with the period from 4am to 11am showing a considerably smaller number of clients.

To further quantify the temporal correlation between the number of clients at various times of the day, we calculate the autocorrelation function for $c(t)$ for various lag values ℓ. Figure 8 shows the results we obtained. It clearly shows the daily periodicity, with peaks around ℓ = 1440, 2880, 4320, ... etc. which are multiples of 1,440 (the number of minutes in a day). The peak correlations also decreases as the lag increases, which is expected.

3.3 Client Interarrival Times

Let $t(i)$ denote the arrival time of the i^{th} session in the trace. Let $a(i) = t(i+1) - t(i)$ denote the interarrival time of the i^{th} and $(i+1)^{\text{th}}$ sessions, where sessions i and $i+1$ belong to different clients. Clearly, $a(i)$ is a time series which describes the interarrival time of clients.

Figure 5 shows the marginal distribution of $a(i)$, which appears to be heavy tailed. In the next section we provide an explanation of this.

[5] We were able to do so for 95% of the IP addresses in our workload.

3.4 Client Arrival Process

The periodic nature of the number of clients observed in the trace over time (Figure 4) suggests that the client arrival process is not stationary. Moreover, Figure 4 (right) and Figure 8 suggest that such non-stationarity is of a periodic nature.

Prior work on characterizing streaming media content [3] suggested that client arrivals were independent, consistent with Poisson arrivals—i.e., exponential interarrivals. In our workload, the client arrival process is *not* stationary in that it is highly dependent on time. That said, it is natural to assume that over a very short time interval, such a process would be stationary, and may indeed be Poisson.

To empirically test this hypothesis, we conducted a simple experiment, in which arrivals were generated using a non-stationary process. This non-stationary process consisted of a sequence of piece-wise-stationary Poisson arrival processes, each of which lasting for 15 minutes. The average arrival rate for each of these stationary Poisson processes was set to reflect the average rates observed in Figure 4 (right). Figure 6 shows the marginal distribution of the resulting interarrival times. The distributions in Figure 5 and in Figure 6 are surprisingly similar,[6] leading us to conclude that a good characterization of the client arrival process is that it is *a piece-wise-stationary Poisson process*, with arrival rates drawn from the periodic patterns shown in Figure 4.

3.5 Client Interest Profile

Over the entire period of the trace, each client (re)visits the live content any number of times. Let k denote the *rank* of a client in terms of the number of requests (or sessions) for that client in the trace. Figure 7 (left) shows the log-log relationship between the number of transfers to (in response to requests from) a client on the Y axis and the rank k of that client on the X axis. Figure 7 (right) shows the log-log relationship between the number of sessions[7] of a client on the Y axis and the rank k of that client on the X axis. These two relationships fit a Zipf-like function (also shown in Figure 7) with $\alpha = 0.7194$ and $\alpha = 0.4704$ ($\pm$ 0.025%), respectively.

[6] The difference between the two distributions seems to be mainly for very large interarrivals. This can be explained by noting that the diurnal mean arrival rate we use to modulate the piece-wise-stationary Poisson process smoothes out the variability in the arrival process. This is evident by comparing the maximum values of the three plots in Figure 4.

[7] Session timeout $T_{\text{off}} = 1,500$ seconds.

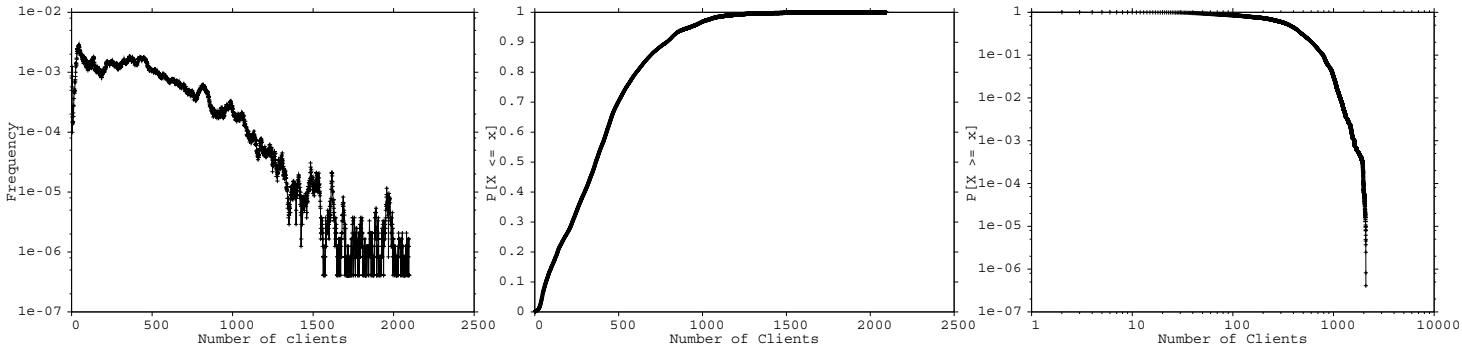

Figure 3: Marginal distribution of number of active clients: Frequency (left), cumulative (center), and CCDF (right)

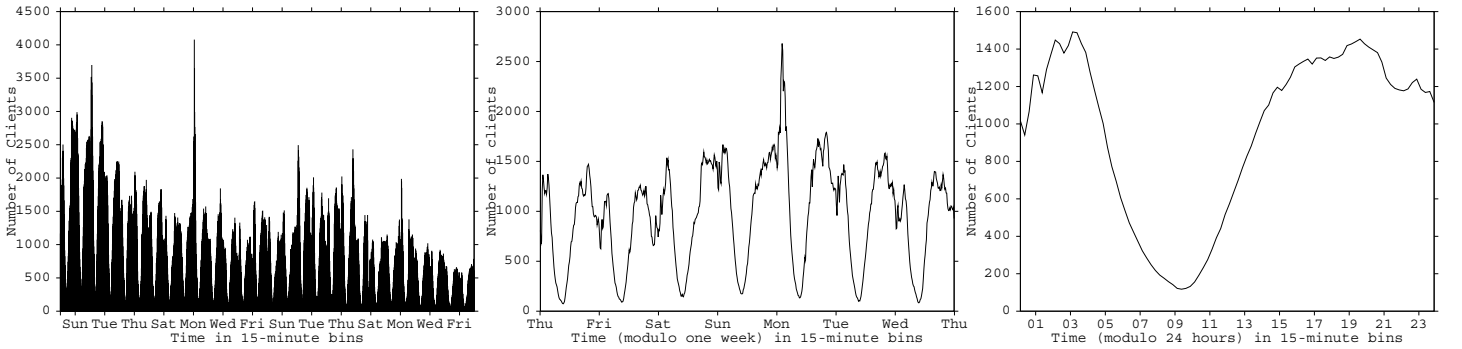

Figure 4: Temporal behavior of number of active clients: Over entire trace duration (left), over week days (center), and hourly (right)

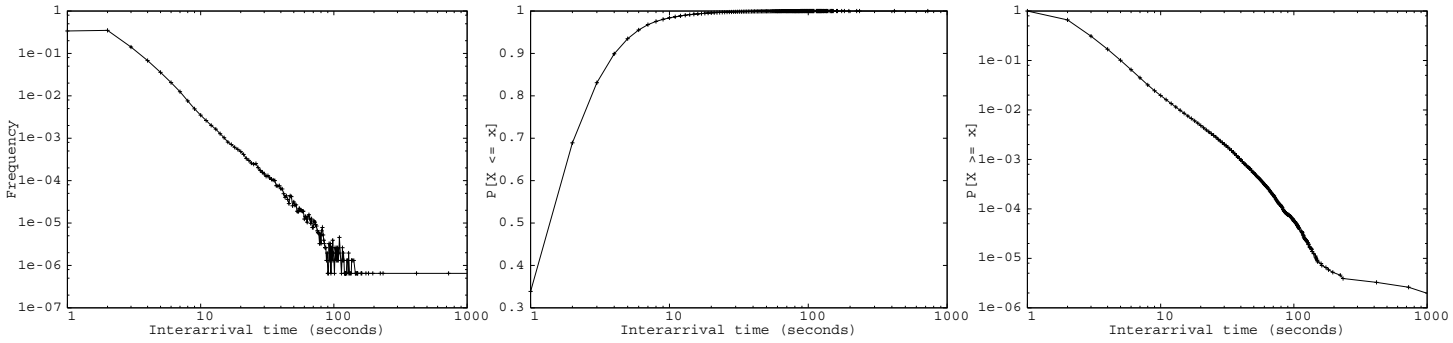

Figure 5: Marginal distribution of client interarrival times: Frequency (left), cumulative (center), and CCDF (right)

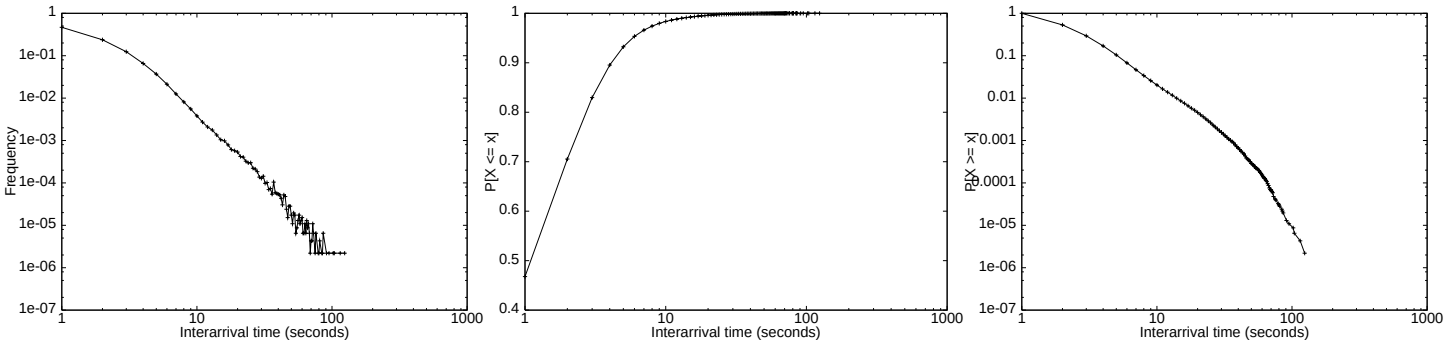

Figure 6: Interarrival times from a piece-wise-stationary Poisson process: Frequency (left), cumulative (center), and CCDF (right)

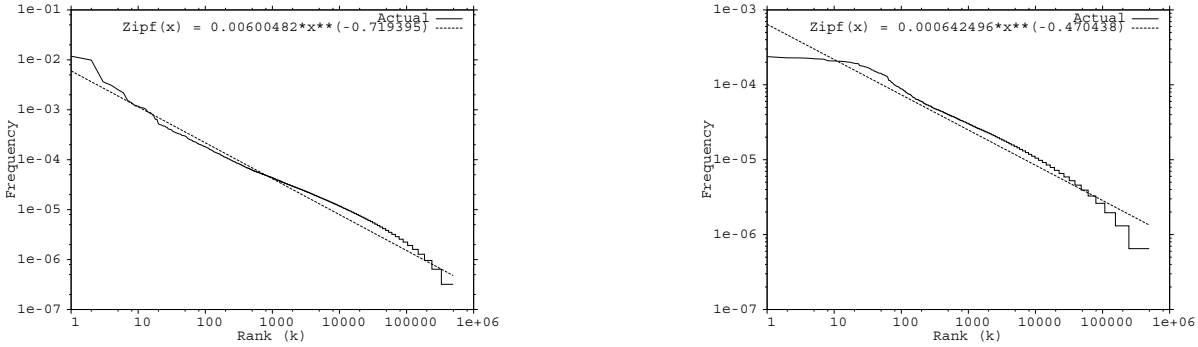

Figure 7: Client Interest Profile: Relationship between client rank and transfer frequency (left) and session frequency (right)

One way of interpreting this relationship is to view the number of requests (or sessions) by a client as a measure of that client's *interest* in the live content. Notice that this notion of interest "inverts" the traditional roles of clients and objects. For stored content delivery (whether pre-recorded streaming media or traditional HTTP file transfers), it is common to think of the *popularity* of a given object (measured in terms of how frequently that object is accessed by various clients). In the context of live content delivery, which is the subject of this paper, characterizing object popularity is not meaningful since clients cannot quite "choose" between objects. Rather, it is more appropriate to gauge the "interest" of a given client in the live content (measured in terms of how frequently that client accesses the various constituent objects of the live content).[8] This role reversal highlights the "duality" of stored versus live media access when it comes to the active versus passive roles of clients and objects.

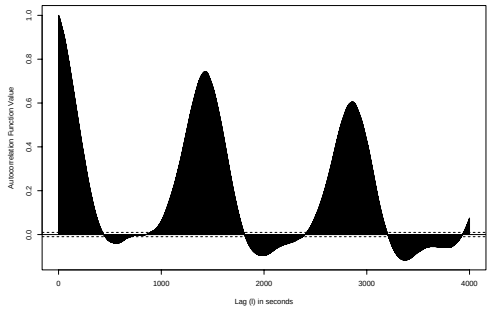

Figure 8: Autocorrelation of number of clients over time

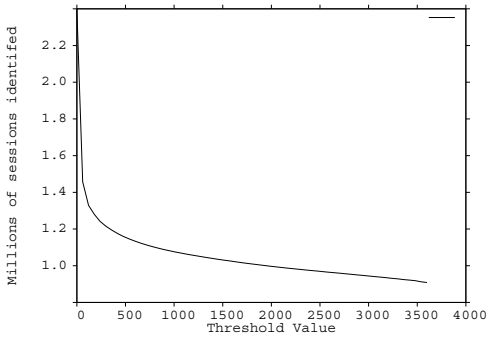

Figure 9: Relationship between number of sessions and T_{off}

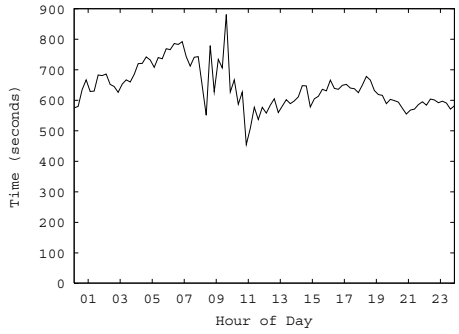

Figure 10: Session ON time versus session starting time

4. Session Layer Characteristics

In this section we present various session characteristics, including session ON/OFF times, as well as correlations between session characteristics and other variables.

4.1 Number of Sessions

Since the trace does not explicitly identify the delimiters of a given session, the number of sessions in the trace depend on our choice of the session timeout parameter T_{off}. Figure 9 shows the relationship between the number of sessions in the trace and the choice of T_{off}. This relationship implies that the number of sessions does not change drastically for $T_{\text{off}} > 1,500$ seconds. For the remainder of this paper, and unless stated otherwise, we use $T_{\text{off}} = 1,500$.

4.2 Session ON Time

Let $l(i)$ denote the length (in seconds) of the i^{th} session in the trace. Clearly, $l(i)$ is the ON time for session i. Figure 11 shows the marginal distribution of $l(i)$ for all sessions identified in the trace. The distribution was fitted to a lognormal distribution with parameters μ = 5.23553 and σ = 1.54432 (also shown in Figure 11).

Figure 11 indicates that session ON times are highly variable. To determine whether this variability is fundamental to the nature of client interactions with live content or whether it is symptomatic

[8] To some extent, client "interest" could be viewed as the popularity of the client as a recipient of content.

of non-stationarity due to temporal correlations (as we discovered for client interarrival times, for example), we characterized the relationship between the length of a session and the time-of-day when the session was started. Figure 10 shows the results of our characterization. It shows a fairly weak correlation between average session length and session starting time. This suggests that the high variability in session length is not due to temporal behaviors (as was the case with client interarrivals), but rather it is a fundamental property of the interaction between users and live content.

4.3 Session OFF Time

Let i and j denote two consecutive sessions in the trace that belong to the *same* client. Let $f(i) = t(j) - t(i) - l(i)$. Clearly, $f(i)$ is the session OFF time (or "log-off" time or "inactive OFF" time). Figure 12 shows the marginal distribution of $f(i)$ for all sessions identified in the trace.

Figure 12 (left) show that large session OFF times seem to form ripples around specific values, which are around 1 day, 2 days, 3 days, *etc*. This underscores the underlying variability in client interests—namely, those "revisiting" the show daily, or every two days, etc. The slight anomaly in the behavior of the distribution for values between 1,500 seconds and 3,000 seconds seems to be the result of a misclassification of OFF times as "session OFF" times as opposed to "transfer OFF" (or "think") times. Recall that our choice of T_{off} = 1,500 is to a large extent arbitrary. As shown in 12 (right), session OFF times fit well an exponential distribution with λ = 203,150 ($\pm$ 0.19%).

4.4 Transfers per Session

Session ON times underscore the continued activity of a given user as reflected by a number of transfers within that session. Figure 13 shows the distribution of the total number of requests (and associated transfers) within each of the sessions identified in the trace. The resulting distribution features a heavy-tailed behavior, which we fitted to a Zipf law with $\alpha = 2.70417$ ($\pm$ 2.7%). We have also studied the correlation between time-of-day and the number of transfers per session, but as was the case for session ON times, we concluded that the variability in the number of transfers per session is not strongly tied to temporal characteristics. Thus, we attribute this variability to the nature of client interactions with live content.

4.5 Interarrivals of Session Transfers

The last variable we characterize at the session layer pertains to the interarrival time between transfers within the same session. Figure 14 shows this distribution, which we fitted to a lognormal distribution with parameters μ = 4.89991 and σ = 1.32074.

5. Transfer Layer Characteristics

In this layer, we are interested in characterizing the workload at the granularity of individual transfers. As we noted earlier, an individual transfer is in response to a specific request by the user. Thus throughout this section, we use the terms "transfers" and "requests" interchangeably.

5.1 Number of Concurrent Transfers

At any point in time t, there are a number of active transfers between the server and some number of clients. This level of concurrency could be used to gauge the load on the server at time t. Figure 15 shows the marginal distribution of the number of concurrent transfers over the entire duration of the trace. Figure 16 (left) shows the mean number of active transfers in intervals of 15 minutes each, over the entire period of the trace. In Figure 4, we also show the periodic behavior of transfers by plotting it over a weekly period (center) and a daily period (right). Not surprisingly, these distributions are fairly similar to those we observed for the number of concurrent clients over time (Figures 3 and 4).

5.2 Transfer Interarrivals

Let $t(j)$ denote the starting time of the j^{th} transfer in the trace. Let $a(j) = t(j+1) - t(j)$ denote the interarrival time of the j^{th} and $(j+1)^{\text{th}}$ transfers. Figure 17 shows the distribution of $a(j)$. The CCDF of $a(j)$ shown in Figure 17 (right) suggests a heavy-tailed nature of that distribution, with two distinct tail behaviors. The first ($\alpha \approx 2.8$) covering interarrivals of up to 100 seconds, and the second ($\alpha \approx 1$) covering interarrivals that are larger than 100 seconds. We argue that these two regimes correspond to two generative processes of client requests, corresponding to transfers during popular time intervals and transfers during unpopular time intervals. We further substantiate this non-stationarity next.

Like client arrivals, the request arrival process is clearly not stationary. In Figure 18, we show the periodic nature of that process by ploting the average request interarrival time over the entire duration of the trace (left), over a revolving weekly period (center), and over a revolving 24-hour period (right). These plots were obtained by computing the average of request interarrival (rounded-up to the closest 1 second) during consecutive 15-minutes periods. While request interarrivals show some variations with respect to the day of the week (e.g., weekends have lower average interarrivals than weekdays), Figure 18 indicates that diurnal behaviors are the main source of variability (with the period from 5am to 11am showing considerably longer interarrivals).

5.3 Transfer Length and Client Stickiness

We now turn our attention to the length of time of individual transfers.[9] Let $l(j)$ denote the length (in seconds[10]) of the j^{th} transfer in the trace. Figure 19 shows the CCDF for $l(j)$ (i.e. Prob[$l(j) > x$]), which we fitted to a lognormal distribution with parameters $\mu = 4.383921$ and $\sigma = 1.427247$.

The size distribution of individual Internet (unicast) transfers has been studied extensively in the literature due to the possible impact that such distribution may have on traffic characteristics. In [14], Crovella and Bestavros argued that the origins of traffic self-similarity can be attributed to the heavy-tailed nature of individual file transfers, which was traced back to the heavy-tailed size distribution of available files. More recent debates [16, 24] as to the true nature of file size distributions (whether Pareto, double Pareto, or Lognormal) further underscore the importance of accurate characterization (and understanding of the root causes) of transfer time distributions.

[9] It is important to note that transfer lengths do not necessarily correspond to transfer ON times since the latter could be the result of overlapped transfers of multiple objects (see Figure 1).

[10] Given the real-time nature of live transmission, the characterization of transfer length in seconds is appropriate. Converting the characteristics to "bytes" would be a function of the transfer rate, which we characterize later.

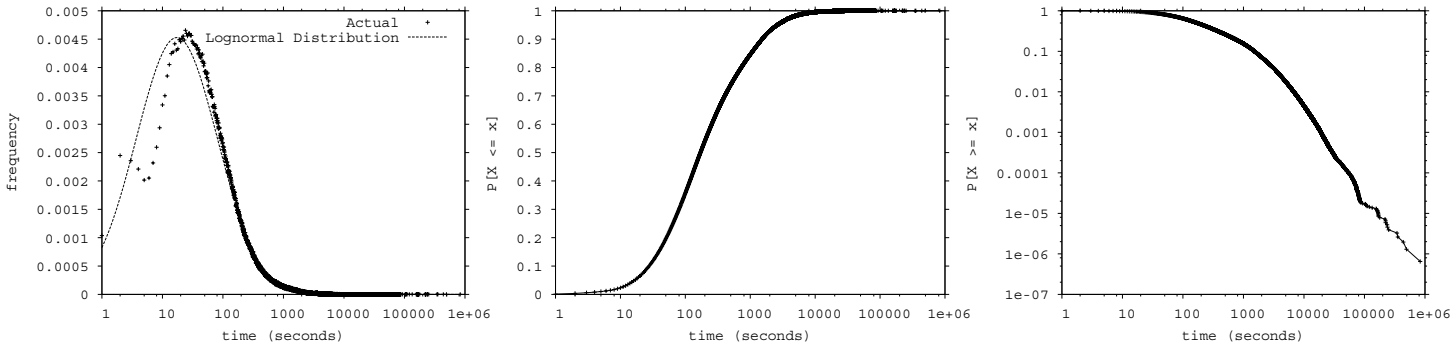

Figure 11: Marginal distribution of session ON times: Frequency fitted to a lognormal (left), cumulative (center), and CCDF (right)

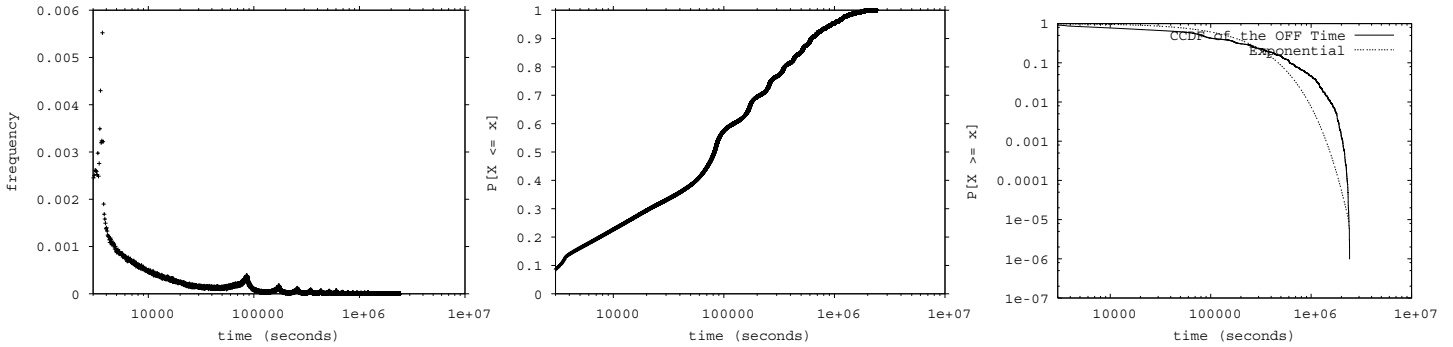

Figure 12: Marginal distribution of session OFF times: Frequency (left), cumulative (center), and CCDF fitted to exponential (right)

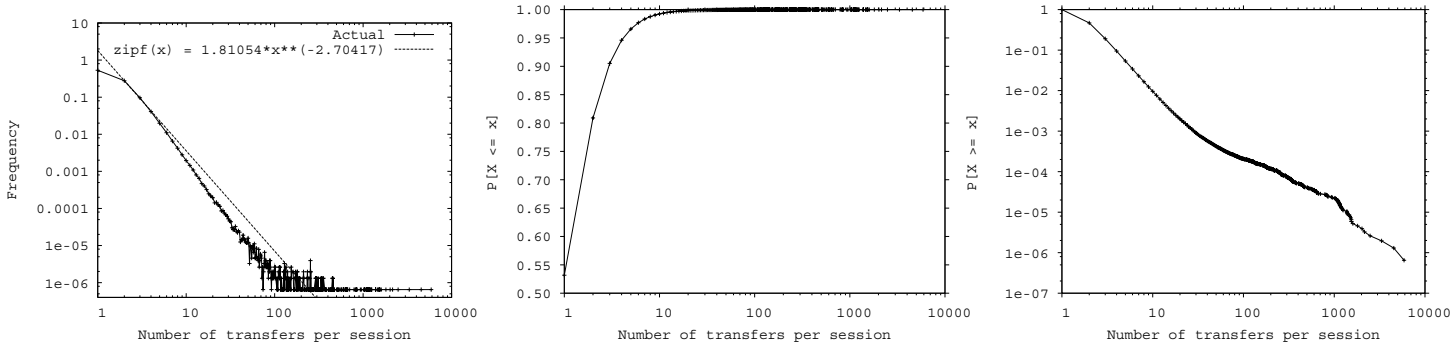

Figure 13: Marginal distribution of number of transfers per session: Frequency (left), cumulative (center), and CCDF (right)

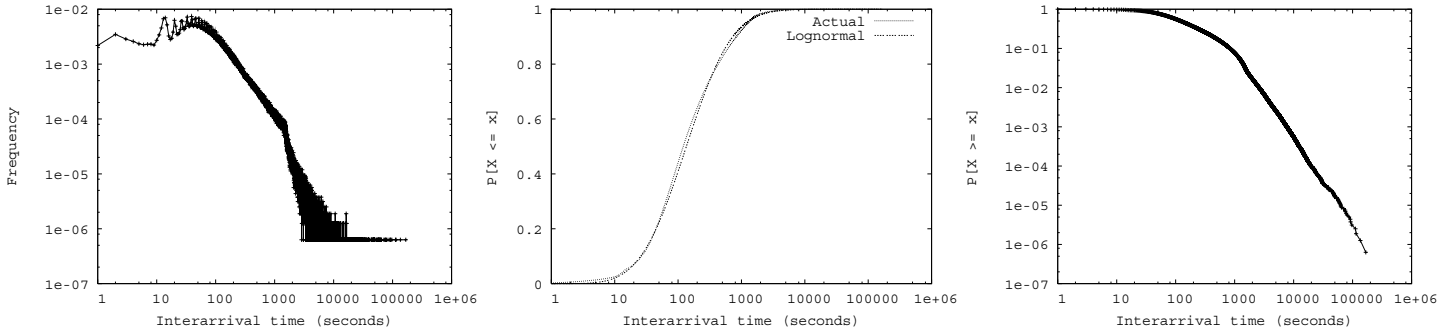

Figure 14: Marginal distribution of transfer interarrivals within a single session: Frequency (left), cumulative fitted to lognormal (center), and CCDF (right)

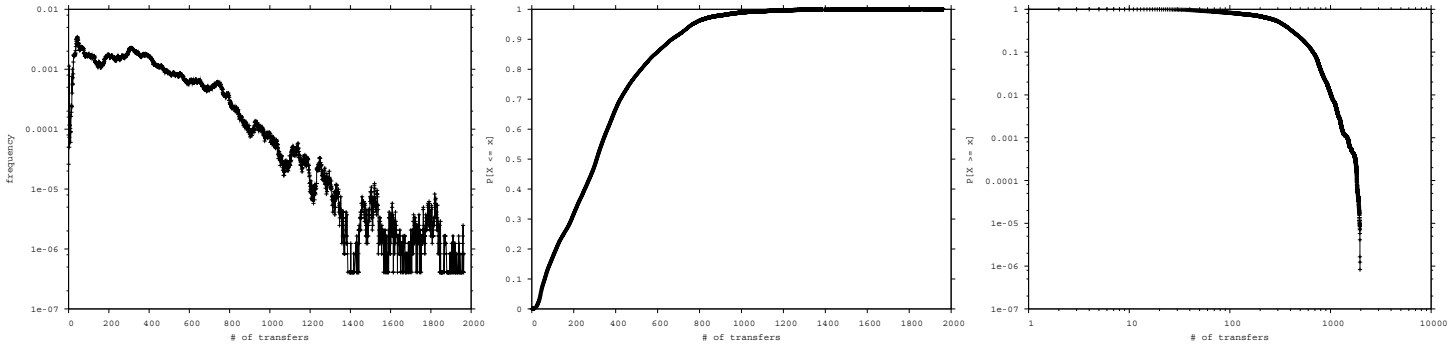

Figure 15: Marginal distribution of concurrent transfers over all sessions: Frequency (left), cumulative (center), and CCDF (right)

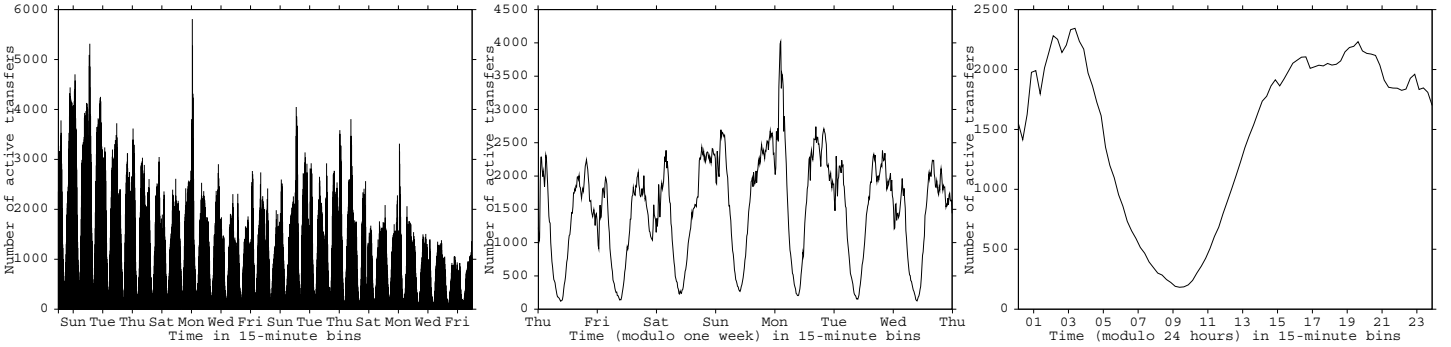

Figure 16: Temporal behavior of number of concurrent transfers: Over entire trace (left), over week days (center), and hourly (right).

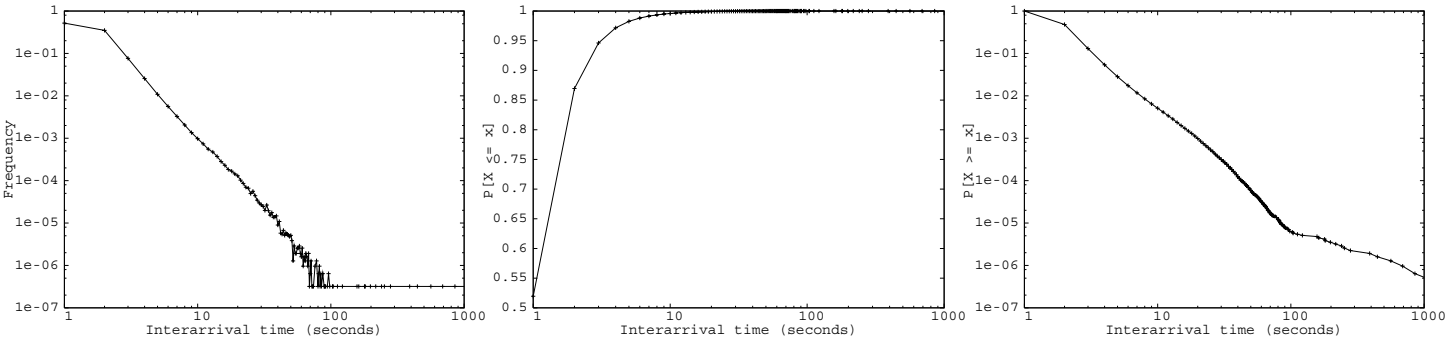

Figure 17: Marginal distribution of transfer interarrival times: Frequency (left), cumulative (center), and CCDF (right)

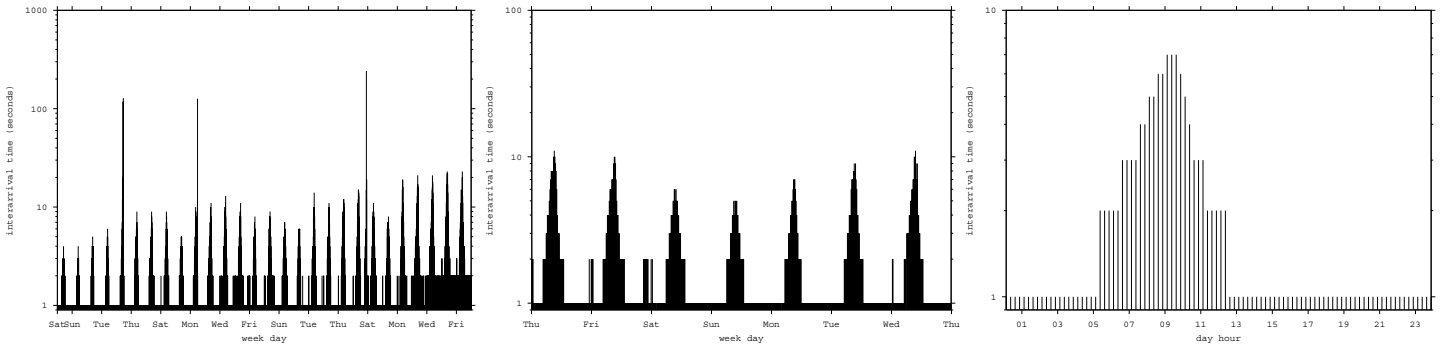

Figure 18: Temporal behavior of transfer interarrival times: Over entire trace (left), over week days (center), and hourly (right).

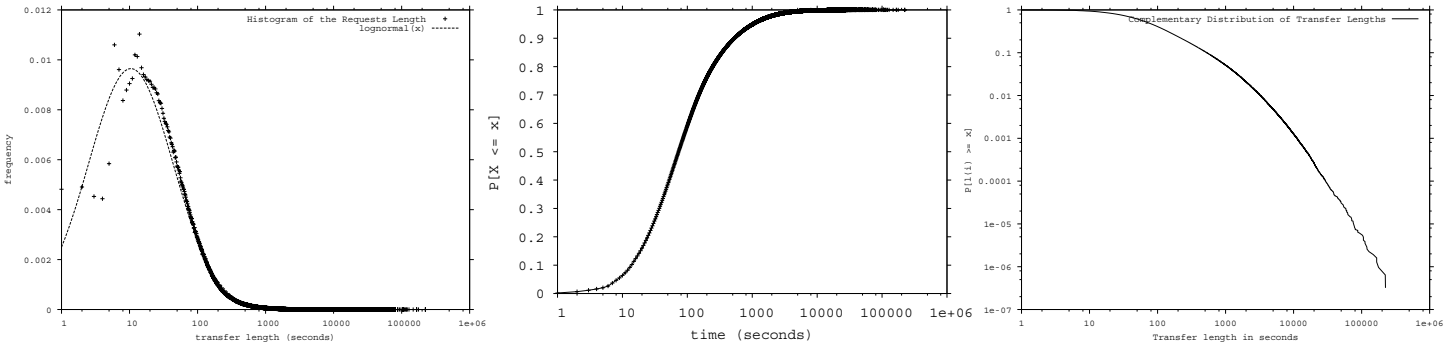

Figure 19: Marginal distribution of transfer lengths: Frequency fitted to lognormal (left), Cumulative (center), and CCDF (right)

For live media content workloads, the long tail of the request ON times is intriguing because it comes about not as a result of available object size distributions, but rather as a result of the client's willingness to "stick" to the live object being transmitted. Recall that for live media, request ON times are bracketed by the start/stop actions performed by clients. Therefore, for live media workloads, the source of high variability in transfer sizes can be traced back to client behavior (as opposed to object size characteristics).

To summarize, for live media workloads, the source of variability in the length of transfers is not due to the classical file size distribution for stored, non-streaming media workloads, but rather to the willingness of a client to "stick" to a transfer.[11]

5.4 Transfer Bandwidth

Figure 20 shows the distribution of bandwidth (in bits per second) experienced by all transfers in the trace. The figure shows two clear "modes". The first is exemplified by the spikes on the right-hand-side of the distribution, which correspond to *client-bound* bandwidth values determined primarily by client connection speeds (e.g., various modem speeds, DSL, cable modem, etc.) The second is exemplified by the much smaller values of bandwidth on the left-hand-side of the distribution, which correspond to *congestion-bound* bandwidth values, resulting from extremely limited network resources.[12]

6. Synthesis of Live Media Workloads

As we discussed earlier, live media workload characterization is crucial to the generation of synthetic (and parametrizable) workloads. In this section, we describe how the results of our hierarchical characterization are used to extend GISMO [19] to generate live media workloads.

6.1 A Generative Model for Live Media Workloads

In our characterization of live streaming media we considered *many* variables at various layers. Many of these variables are not independent. For example, the client interarrival time distribution follows from the distribution of the number of clients and the distribution of session ON and OFF times. Having some redundancy in the characterization is fine as it helps us understand various nuances of the access patterns. But when it comes to using the results of a characterization to generate synthetic workloads, we have to make choices as to which variables are to be used to generate the synthetic trace. Such choices are made based on an explicable *generative model*. In this section, we present such a model, along with the subset of variables (from our characterization in the previous sections) that are necessary for model instantiation.[13]

Our model for synthetic workload generation consists of the following ingredients, which are loosely associated with the three layers of our characterization hierarchy.

Client Arrivals:. To be able to generate sessions (and eventually transfers within these sessions), we must determine *when* these sessions are started and *which* clients initiate them. To determine *when* client arrivals occur, we use a non-stationary Poisson process whose mean is keyed to the periodic behavior of Figure 4. To determine *which* client should be associated with a given arrival, we use the client interest profile of Figure 7 (right).

Session Length:. The arrival of a client underscores the start of a session. To be able to generate transfers within that session, we need to determine *how many* such transfers to generate. This is determined using the distribution in Figure 13.

Transfers:. To generate transfers within a specific session, we need to determine *when* each transfer starts, and *how long* each transfer ought to be. By definition, we note that the first transfer starts with the session arrival time. The start time of the following transfers in the session (if any) could be determined using the distribution of the interarrival time of intra-session transfers in Figure 14. The length of each transfer is determined using the distribution of transfer lengths shown in Figure 19.

Table 2 summarizes the subset of variables we retained in our generative model, as well as the specific distributional properties of these variables as suggested by our characterization of the workload at hand.

It is important to note that, as we surmised at the outset, many of the characteristics of live media workloads are likely to depend heavily on the application at hand—e.g., the periodicity observed in our reality TV application is likely to be very different from that observed in (say) live feeds associated with a soccer game. That said,

[11] It is important to note that for stored streaming content, *both* object size and client interactivity play a role in the length of transfers.

[12] Figure 20 (righ) suggest that around 10% of all transfers were congestion-bound. Notice that server overload conditions are *not* a factor as we discussed in Section 2.4.

[13] It is important to note that our generative model is *not* unique. Indeed, we have toyed with other models, but decided on the model presented in this section for its explicative appeal.

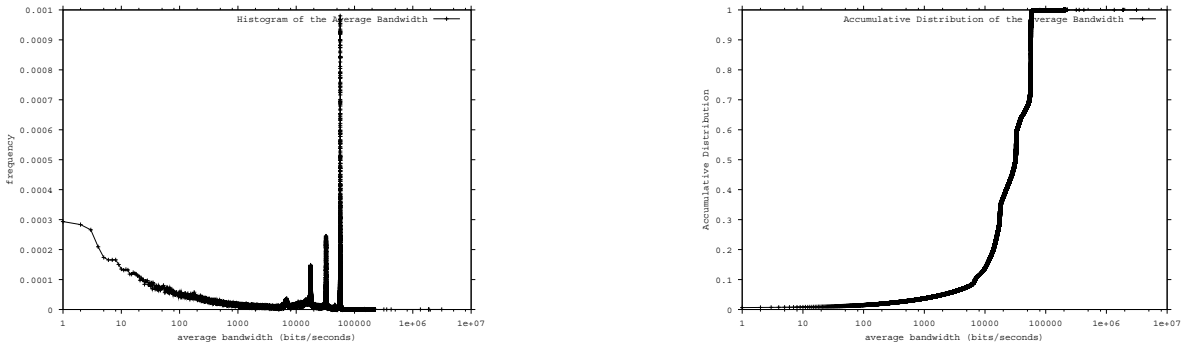

Figure 20: Transfer bandwidth: Frequency (left) and Cumulative distribution (right)

Variable	Distribution	Parameters / Settings	Source
Mean Client Arrival Rate $f(t)$	Periodic over p	p = 24 hours	Figure 4
Client Arrival Process	Piece-wise-stationary Poisson	$\lambda = f(t)$	Figure 5
Client Interest Profile	Zipf	α = 0.4704	Figure 7
Transfers per Session	Zipf	α = 2.7042	Figure 13
Interarrival of Session Transfers	Lognormal	$\mu = 4.900$, $\sigma = 1.321$	Figure 14
Transfer Length	Lognormal	$\mu = 4.384$, $\sigma = 1.427$	Figure 19

Table 2: Summary of the variables retained for the synthesis of live streaming media workloads in GISMO

we believe that the generative processes we described here can be easily adjusted to specific distributions associated with other applications. Indeed, this is one of the features of the GISMO framework we use to synthetically generate streaming media workloads [19].

6.2 GISMO Extensions

GISMO (a Generator of Internet Streaming Media Objects and workloads) is a toolset that enables the synthesis of streaming access workloads. GISMO was initially aimed at generating pre-recorded media objects (such as video and new clips) and workloads. As such, it enables the generation of synthetic workloads, which are parameterized so as to match properties observed in real workloads, including object popularity, temporal correlation of requests, client session length, seasonal access patterns, client VCR inter-activities, and self-similar variable bit-rate.

A workload generated by GISMO consists of a set of objects (with popularity distribution, size distribution, and variable bit-rate content encoding), and a sequence of user sessions (with possibly inter-activities within each session). Although many of these characteristics are still applicable to the synthesis of live media workloads (e.g., VBR characteristics of content), we found it necessary to extend GISMO to enable us to capture the fundamental difference between pre-recorded and live media workloads—namely the role reversal of clients and objects. We give two specific examples below.

From our characterization of the client arrival process, it is clear that client arrivals are highly correlated. This requires us to introduce the notion of non-stationary of arrivals in GISMO. We do so by allowing the parameters of the arrival processes to be programmable (e.g., by calling a user-supplied function reflecting diurnal patterns, for example).

From our analysis of client interests in the live content, we concluded that there is a significant Zipf-like skew in the frequency of access across the client population. To reflect this in GISMO synthetic traces required us to introduce clients as unique entities, and to allow the association of sessions to clients to follow a particular distribution (e.g., Zipf). Notice that this added feature (of associating a client to a GISMO session) is analogous to the existing feature (of associating an object to a GISMO session). In a sense, our modification of GISMO allows *both* ends of a session to be selected preferentially from amongst an enumerable set of clients and objects to reflect object popularity and/or client interest profiles.

7. Related work

Workload characterization is fundamental to the synthesis of realistic workloads. Many studies focused on the characterization and generation of non-streaming (such as HTTP) workloads (e.g., [4, 5, 6, 8, 7, 9, 14, 15, 17, 28, 27]). These studies have improved our understanding of the nature of access patterns involving stored, non-streamed content (e.g., documents). Some of the important findings of these studies include the characterization of Zipf-like document popularity distributions, heavy-tailed object and request size distributions, and reference locality properties. A discussion of the various characteristics of workloads involving non-streamed content (while relevant to some aspects of our work) is outside the scope of this paper. Thus, in the remainder of this section, we restrict our coverage of related work to studies of streaming media workload characterization and synthesis.

Streaming Media Access Characterization:. Several previous studies [26, 18, 2, 11, 3], have characterized workloads of pre-recorded media object access primarily from media servers for educational purposes. We summarize these efforts below.

Padhye and Kurose [26] studied the patterns of user interactions with a media server in the MANIC system. They characterized session length and user activity within a session. A session was

considered a sequence of alternating ON periods (when the user is retrieving the media) and OFF periods (when no media is being streamed to the user). The distributions of both ON period and OFF period appeared to be heavy-tailed—i.e., lognormal or gamma distributions. They also observed user jumps and "locality" in the jumps.

Acharya and Smith characterized user access to video objects on the Web [2]. Their analysis revealed the existence of strong temporal locality of reference. Accesses exhibited geographical locality—i.e., a small number of local machines accounted for most of the requests. They observed skewed popularity of video objects, which did not follow a Zipf distribution. In addition, nearly a half of the requests were for a partial access of the object, indicating early stoppage of transfers by users.

Chesire et al. [11] analyzed a client-based streaming-media workload collected from the border routers serving the University of Washington. The work focused on the characterization of object size, server and object popularity, session statistics, sharing patterns, and bandwidth utilization. They found that most streaming objects are small. However, they also found that a small percentage of requests were responsible for almost half of the total bytes. The popularity of objects was found to follow a Zipf-like distribution. They also observed that requests during the periods of peak loads exhibited a high degree of temporal locality. Using this workload, they also studied the effectiveness of caching and multicast for reducing the bandwidth requirements of streaming media delivery.

Almeida el al. [3] analyzed workloads from two media servers for educational purposes. During periods of approximately stationary request arrival rates, the client session arrival process was found to be approximately Poisson, and the time between interactive requests followed a Pareto distribution. The popularity of the media objects they considered can be modeled by the concatenation of two Zipf-like distributions. They found that the segments of media objects are not accessed equally frequently; for less popular objects, the earlier segments are more likely to be accessed. The distribution of delivered media per session (or per request within a session) was found to depend on the object's length. For long objects, this distribution was often heavy-tailed. Also, they uncovered a high degree of user interactivity in the workload, which implied that the effectiveness of multicast delivery is limited.

Streaming Traffic Characterization:. Several studies [21, 10, 20, 29] have focused on low-level dynamics of streaming access, such as packet loss and delay, network transport protocols.

Mena and Heidemann [21] examined the traffic emanating from a popular Internet audio service using the RealAudio program. They found a pervasive use of non-TCP friendly transport protocols, and strong consistencies in audio traffic packet sizes and data rate patterns. Recently, based on this study, Lan and Heidemann [10] identified the structural properties of RealAudio traffic, and developed and validated an application-level simulation model.

Loguinov and Radha [20] analyzed several network performance metrics including packet loss, round-trip delay, one-way delay jitter, packet reordering, and path asymmetry. In particular, their findings suggest that Internet packet loss is bursty. Both the distributions of loss burst length and round-trip time appear to be heavy-tailed.

Wang, Claypool, and Zuo [29] analyzed RealVideo traffic from several Internet servers to many geographically diverse users. They mainly focused on frame rate and the influence of client-side bandwidth. They found that typical RealVideos achieve a reasonably high quality (average frame rate of 10 frames per second and higher). Video performance is most influenced by the bandwidth of the end-user connection to the Internet, but high-bandwidth Internet connections are pushing the video performance bottleneck closer to the server.

8. Summary and Conclusion

In this paper we have presented what we believe to be the first characterization of *live* streaming media delivery on the Internet. Our characterization adopted a hierarchical approach at three layers, corresponding to clients, sessions, and transfers. Our characterization has uncovered a number of interesting observations, in each of these layers.

Client Layer:.

- The arrival process of clients can be modeled by a piece-wise stationary Poisson process, which is characterized by (1) a strong diurnal pattern that determines the average arrival rate over consecutive intervals of time, and (2) Poisson arrivals with the preset average rate for each interval.
- The identity of the client making a request can be modeled by a skewed Zipf-like distribution.

Session Layer:.

- The session ON time follows approximately a Lognormal distribution, and does not appear to be as heavy as Pareto.
- The session OFF time follows approximately an exponential distribution.
- The number of transfers within a session appears to be skewed and can be modeled by a Zipf distribution.

Transfer Layer:.

- The transfer arrival process exhibits properties similar to the client arrival process (and hence the same generative process we devised could be used).
- Transfer lengths, which are attributed to client stickness, follows approximately a Lognormal distribution, which is consistent with the session ON time distribution.
- Transfer bandwidth is primarily determined by client connection speeds, with approximately 10% of the transfers being severely limited by limited network resources.

Characteristics of live media access patterns are significantly different from those of traditional stored object workloads, whether streamed (e.g., pre-recorded media clips) or not (e.g., files). The difference stems from the role reversal of objects and clients in live versus stored content delivery environments. Accesses to stored objects are *user driven*, whereas accesses to live objects are *object driven*. This observation, together with the results of our hierarchical characterization, helped us enhance the GISMO toolset to generate realistic live media workloads.

In this paper, we did not characterize the properties of the network as reflected in the logs we analyzed. Also, we did not study the impact that network congestion, as reflected by increased packet drops or lost connections would have on user access patterns. We are currently investigating these issues.

Acknowledgments

The authors would like to thank the anonymous Brazilian Reality Show web site owners and operators for enabling this research to proceed by providing us access to their logs.

References

[1] Soam Acharya and Brian Smith. An experiment to characterize videos stored on the Web. In *Proceedings of MMCN*, 1998.

[2] Soam Acharya, Brian Smith, and Peter Parns. Characterizing user access to video on the World Wide Web. In *Proceedings of MMCN*, January 2000.

[3] Jussara Almeida, Jeffrey Krueger, Derek Eager, and Mary Vernon. Analysis of educational media server workloads. In *Proceedings of NOSSDAV*, June 2001.

[4] Virgilio Almeida, Azer Bestavros, Mark Crovella, and Adriana de Oliveira. Characterizing reference locality in the WWW. In *Proceedings of PDIS*, December 1996.

[5] Martin Arlitt and Carey Williamson. Web server workload characteristics: The search for invariants. In *Proceedings of SIGMETRICS*, May 1996.

[6] Gaurav Banga and Peter Druschel. Measuring the capacity of a Web server. In *Proceedings of USITS*, December 1997.

[7] Paul Barford, Azer Bestavros, Adam Bradley, and Mark Crovella. Changes in Web client access patterns: Characteristics and caching implications. *World Wide Web*, 2(1):15–28, 1999.

[8] Paul Barford and Mark Crovella. Generating representative Web workloads for network and server performance evaluation. In *Proceedings of SIGMETRICS*, June 1998.

[9] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of INFOCOM*, April 1999.

[10] Kun chan Lan and John Heidemann. Multi-scale validation of structural models of audio traffic. Technical Report ISI-TR-544, USC Information Sciences Institute, 2001.

[11] Maureen Chesire, Alec Wolman, Geoff Voelker, and Henry Levy. Measurement and analysis of a streaming workload. In *Proceedings of USITS*, March 2001.

[12] CNN. The CNN NewsPass Subscription Service. http://www.cnn.com.

[13] Microsoft Corporation. Windows Media Services 4.1. http://www.microsoft.com/windows/windowsmedia /technologies/services.asp.

[14] Mark Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceedings of SIGMETRICS*, May 1996.

[15] Carlos Cunha, Azer Bestavros, and Mark Crovella. Characteristics of WWW client-based traces. Technical Report BU-CS-95-010, Computer Science Department, Boston University, April 1995.

[16] Allen B. Downey. The structural cause of file size distributions. In *Proceedings of International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, August 2001.

[17] Steven D. Gribble and Eric A. Brewer. System design issues for Internet middleware services: Deductions from a large client trace. In *Proceedings of USITS*, December 1997.

[18] Nissim Harel, Vivekanand Vellanki, Ann Chervenak, Gregory Abowd, and Umakishore Ramachandran. Workload of a media-enhanced classroom server. In *Proceedings of Workshop on Workload Characterization*, 1999.

[19] Shudong Jin and Azer Bestavros. GISMO: Generator of Streaming Media Objects and Workloads. *Performance Evaluation Review*, 29(3), 2001.

[20] D. Loguinov and H. Radha. Measurement study of low-bitrate internet video streaming. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2001.

[21] Art Mena and John Heidemann. An empirical study of real audio traffic. In *Proceedings of Infocom*, March 2000.

[22] D. A. Menascé and V. A. F. Almeida. *Capacity Planning for Web Services: metrics, models, and methods*. Prentice Hall, Upper Saddle River, NJ, 2002.

[23] D. A. Menascé, V. A. F. Almeida, R. Riedi, F. Pelegrinelli, R. Fonseca, and W. Meira Jr. In search of invariants for e-business workloads. In *Proceedings of the 2000 ACM Conference in E-commerce*, October 2000.

[24] Michael Mitzenmacher. Dynamic models for file sizes and double pareto distributions, 2002. Preprint.

[25] Real Networks. The RealONE SuperPass Subscription Service. http://www.real.com.

[26] J. Padhye and J. Kurose. An empirical study of client interactions with a continuous-media courseware server. In *Proceedings of NOSSDAV*, June 1998.

[27] Venkata N. Padmanabhan and Lili Qiu. The content and access dynamics of a busy Web site: Findings and implications. In *Proceedings of SIGCOMM*, August 2000.

[28] V. Paxson. Wide-area traffic: The failure of poisson modeling. In *Proceedings of SIGCOMM*, August 1994.

[29] Yubing Wang, Mark Claypool, and Zheng Zuo. An empirical study of video performance across the internet. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2001.