A Hierarchical Characterization of a Live Streaming Media Workload

Eveline Veloso, Virgílio Almeida, Wagner Meira, Jr., Azer Bestavros, *Member, IEEE*, and Shudong Jin, *Member, IEEE*

***Abstract*—We present a thorough characterization of what we believe to be the first significant *live* Internet streaming media workload in the scientific literature. Our characterization of over 3.5 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 in contrast 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 client stickiness to a particular live object, as opposed to structural (size) properties of objects. Second, by contrasting two live streaming workloads from two radically different applications, we conjecture that some characteristics of live media access workloads are likely to be highly dependent on the nature of the live content being accessed. This dependence is clear from the strong temporal correlation observed in the traces, which we attribute to the impact of synchronous access to live content. Based on our analysis, we present a model for live media workload generation that incorporates many of our findings, and which we implement in GISMO.**

***Index Terms*—Internet, live streaming, measurement, multimedia, workload characterization.**

I. Introduction

THE use of the Internet as a channel for the delivery of streaming media content such as video and audio 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]–[3], [11], [24], [29]. However, to the best of our knowledge, all these studies targeted pre-recorded, stored streams (e.g., news clips, film trailers, educational clips) and none has considered the characterization of *live* streams (e.g., camera feeds). This paper provides such a characterization for a unique data set capturing hundreds of thousands of live streaming 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). 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.).

While workload characterization is an important ingredient of performance evaluation and prediction in general, it is particularly critical for proper capacity planning of live 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 under-provisioned 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. Note that many content providers are now charging for access to streaming content, e.g., CNN's NewsPass [12] and Real Networks' RealOne SuperPass [28] subscription services. Capacity planning based on accurate understanding of workload characteristics [25] becomes a necessity. A case in point is the experience of thousands of users in January 1999 when attempting to view VictoriasSecret.com's highly advertised webcast.

The characteristics of live streaming workloads are likely to be fundamentally different from those of pre-recorded, stored clips [33], [34]. 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

Manuscript received October 17, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Rexford. The work of A. Bestavros was supported in part by the National Science Foundation (NSF) under Research Grants ANI-9986397, ANI-0095988, ANI-0205294, EIA-0202067, 0524477, and 0520166. The work of V. Almeida, W. Meira, Jr., and E. Veloso was supported in part by a number of research grants from CNPq-Brazil.

E. Veloso, V. Almeida, and W. Meira, Jr. are with the Computer Science Department, Federal University of Minas Gerais, Belo Horizonte 31270-010, Brazil (e-mail: eveline@dcc.ufmg.br; virgilio@dcc.ufmg.br; meira@dcc.ufmg.br).

A. Bestavros is with the Computer Science Department, Boston University, Boston, MA 02215 USA (e-mail: best@cs.bu.edu).

S. Jin is with the Computer Science Division, Case Western Reserve University, Clevveland, OH 44106 USA (e-mail: jins@case.edu).

Digital Object Identifier 10.1109/TNET.2005.863709

operations possible with stored media (e.g., VCR functions) are simply not available for live media. More importantly, the correlation 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; 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.

These 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 *content driven*; they are directly influenced by aspects related to the nature of the object, e.g., show/event 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 the streams they access: they can only join or leave the audience of the live "active" content. Notice that we do not consider synchronous rebroadcast of pre-recorded content to constitute "live" content. While the synchronous nature of such rebroadcasts is likely to make their characteristics different from those of asynchronously accessed stored content, we argue that "liveness" is an attribute that encompasses "synchrony" (the difference between a movie premier and a pay-per-view rebroadcast of the movie).

The remainder of this paper is organized as follows. In Section II, we describe the source of the logs used in this research. 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 Section III), session characteristics (in Section IV), and object request characteristics (in Section V). We have extended GISMO [21], 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 VII. In Section VIII, we present an overview of related work. We conclude in Section IX with a summary of our findings.

II. Live Streaming Workload

A. Source of the Workload

We obtained logs of over one month 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 distinct 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 + video) feeds captured from one of 48 different cameras embedded in the environment surrounding the contestants in the reality show.

B. 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 [26]. 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 are grouped into sessions. At the top level, sessions from individual clients are grouped into a client behavior level.

Throughout this paper, we use the term *live streams* (or simply *streams* when liveness is clear from the context) to refer to "continuous" feeds whose existence is defined by the duration of an event (e.g., live show or game). We characterize access to such streams 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 correlation).

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 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 streams 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). Fig. 1 shows how client activities (namely request start/stop) translate to various session ON and OFF times. In particular, a session is the period of time during which the transfer of content to the client is not stopped for more than a given threshold T_{off}.

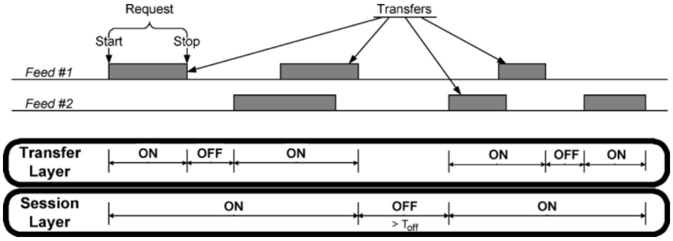

Fig. 1. Relationship between client activities and ON/OFF times at the session and transfer layers.

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 streams, a transfer is the result of a pair of requests to "start" and eventually "stop" viewing a live feed. For stored video, other requests may include VCR functionalities (e.g., "pause", "fast-forward", "rewind", etc.) 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 streams (e.g., different live feeds). During transfer OFF times (which by definition must be smaller than T_{off}) no live streams are served to the client. Transfer OFF times correspond loosely to "think" times or to what has been termed "active OFF" times in [15]. Fig. 1 shows how client activities (start and stop requests) 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, and 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 *streams*. 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 VII).

C. Log Statistics and Server Configuration

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

TABLE I
BASIC STATISTICS OF THE TRACE USED IN THIS PAPER

Log period	28 days in early 2002
Total # of live streams	2
Total # of client ASs	1,010
Total # of client IPs	364,184
Total # of users	691,889
Total # of sessions	1,422,021
Total # of transfers	3,172,486
Total content served	4.65 TeraBytes

While the Windows Media Server supports both unicast and multicast services, only unicast transfers were enabled. For each one of the two streams it served, the Windows Media Server provided four distinct encodings to match various client bandwidth profiles (e.g., modem versus DSL). Unfortunately we do not have direct knowledge of the settings for these different resolutions. However, our empirical measurement of the bit-rate of individual transfers suggests that these four encodings corresponded to bit rates that are roughly around 7 kb/s, 18 kb/s, 32 kb/s, and 57 kb/s.

The Windows Media Server was configured to enable full logging of all user activities throughout the log collection period. Each entry in the log identifies a single client/server request/response. For each entry in the log, the following information is provided:

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 stream;
4) Transfer statistics, e.g., average bandwidth;
5) Server load statistics, e.g., server CPU utilization;
6) Other information, e.g., URL, HTTP status;
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.

Log Sanitization: We have identified a number of problems with a small percentage of the entries in the logs we used.[1] These requests were excluded from our characterization.

D. Fitting Procedures

Throughout this paper, we model various aspects of the workload using distributions which we fit to the empirical data we obtained from the logs. Unless we mention otherwise, all our fitted distributions yielded a correlation coefficient that well exceeds 0.95.

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., several thousands). To ensure that the characteristics we present throughout the paper are not affected by server overload, we have analyzed the logs and indeed established that periods of server overload 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.

III. 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.

[1] Specifically, these entries had erroneous timestamps (e.g., resulting in user sessions spanning durations longer than the 90-day period of the show!) They were all traced to a (perhaps misconfigured or buggy) MacOS client.

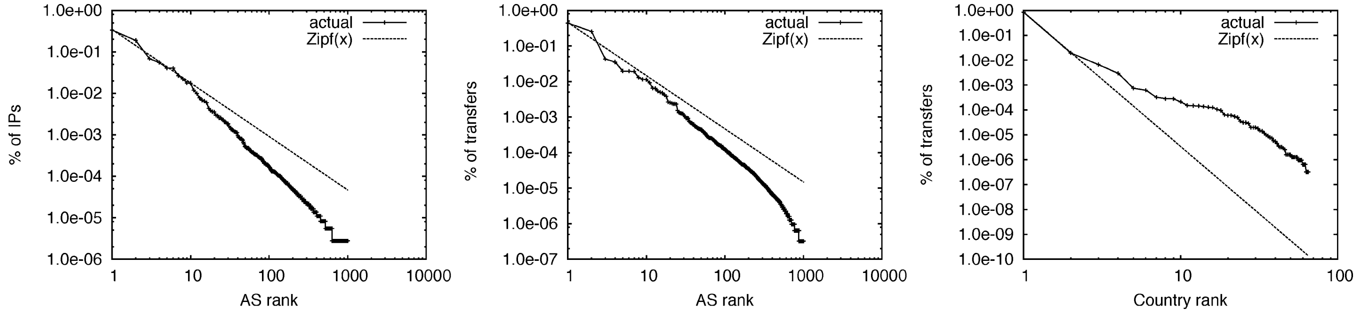

Fig. 2. Client diversity: IP addresses over ASs (left), transfers over ASs (center), and transfers over countries (right).

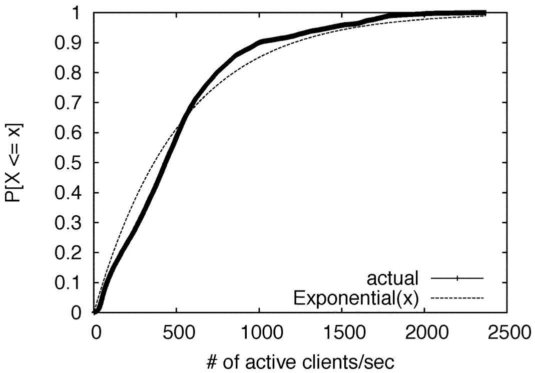

Fig. 3. Cumulative distribution of number of active clients.

A. Topological and Geographical Distribution of Client Population

An important question that is often asked regarding workload characterization studies has to do with the "significance" of the logs underlying the characterization. As evident from Table I, 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), the number of accesses made by these clients, as well as the diversity of the population. Using IPAS [17], a software package from NLANR for IP address to Autonomous System (AS) conversion, we translated client IP addresses to AS numbers, which in turn were mapped to countries using conversion tables published by CAIDA [18]. We were able to do so for 95% of the IP addresses in our workload. Our mappings identified over 1000 different ASs scattered over 65 countries. Fig. 2 shows the "popularity" of each AS in our workload as measured by the number of IP addresses (left) and by the number of transfers (center) that have been traced back to that AS, respectively. Fig. 2 (right) shows the distribution of transfers over the various countries. All three plots suggest a Zipf-like profile, with parameter $\alpha = 1.29, 1.49,$ and 5.4, respectively.

B. Client Concurrency Profile

At any point of time t, a number of clients $c(t)$ are considered active, in the sense that their sessions are still in progress. This level of concurrency could be used to gauge the popularity of the particular content being transmitted at time t. Fig. 3 shows the marginal distribution of $c(t)$ over the entire trace (measured over 15-minute intervals or bins).

Notice that many factors may contribute to the wide variability observed in the number of concurrently active clients. These include activities occurring 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). Fig. 4 (left) shows the average value of $c(t)$ calculated for consecutive 15-minute bins, over the entire trace. Also, in Fig. 4, we show the periodic behavior of $c(t)$ by plotting $c(t \oplus p)$, 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), Fig. 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. Fig. 5 shows the results. It clearly shows the daily periodicity, with peaks around $\ell = 1440, 2880, 4320, \ldots$ etc. which are multiples of 1440 (the number of minutes in a day). The peak correlation also decreases as the lag increases, which is expected.

C. Client Interarrival Times

To characterize client interarrival times (IAT), we utilize a time series $t(i)$ to denote the arrival time of the ith session in the trace. The time series $a(i)$ is defined as $t(i+1) - t(i)$ and it denotes the interarrival time of the ith and $(i+1)$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.

Fig. 6 shows the frequency (left) and CCDF (center) distributions of $a(i)$, which we fitted to a Pareto distribution ab^a/x^{a+1}, with parameters $a = 2.52$ and $b = 1.55$ for $x < 200$ seconds, and with parameters $a = 0.76$ and $b = 2.1e - 05$ for $x > 200$.

The periodic nature of the number of clients observed in the trace over time (Fig. 4) suggests that the client arrival process is not stationary. Moreover, Fig. 4 (right) and Fig. 5 suggest that such nonstationarity 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. This is consistent with findings in other settings (e.g., arrival processes for the same Web document [5], and for telnet and FTP sessions [32]).

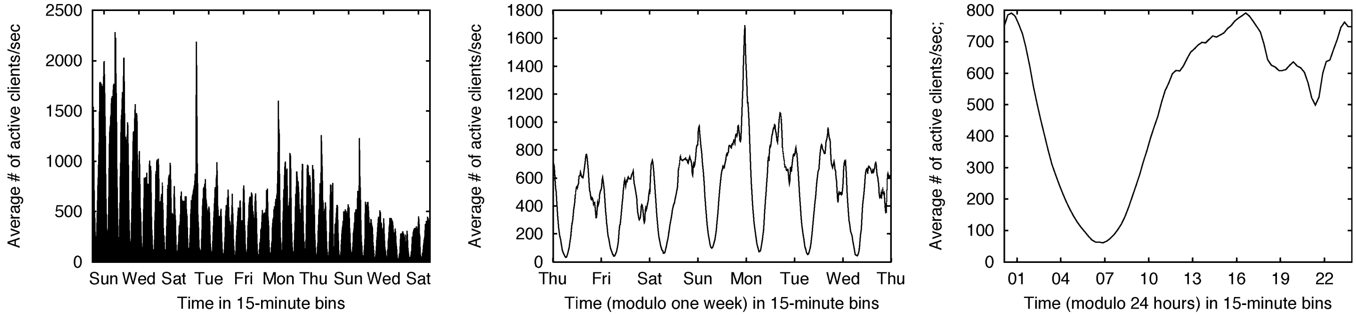

Fig. 4. Temporal behavior of number of active clients: over entire trace (left), daily (center), and hourly (right).

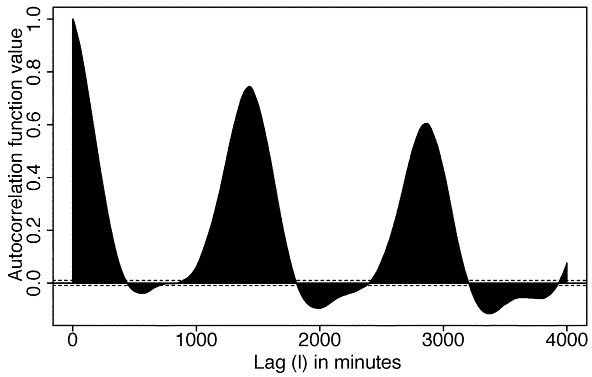

Fig. 5. Autocorrelation of number of clients over time.

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 nonstationary process. This nonstationary 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 Fig. 4 (right). Fig. 6 (right) shows the frequency distribution of the resulting interarrival times. The distributions showed in Fig. 6 (left) and (right) are surprisingly similar,[2] 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 Fig. 4. To gauge the "stationarity" of this process, we repeated this experiment with various periods of stationarity Poisson arrival rates (1 minute, 5 minutes, 15 minutes, 1 hour, etc.). Our findings show that stationarity periods larger than one hour produced a marginal distribution that diverged significantly from that in Fig. 6 (left), leading us to conclude that the arrival process could be assumed stationary at time scales of dozens of minutes.

D. Client Interest Profile

Over the entire trace, each client visits the live content any number of times, indicating some level of interest in the live content of the stream. To characterize the interest profile of the client population, let k denote the *rank* of a client in terms of the number of requests (or sessions) for that client. Fig. 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 (based on number of requests from that client relative to all other clients) on the X axis. Fig. 7 (right) shows the log-log relationship between the number of sessions of a client on the Y axis and the rank k of that client (based on number of sessions from that client relative to all other clients) on the X axis. These two relationships fit a Zipf-like function (also shown in Fig. 7) with $\alpha = 0.719$ and $\alpha = 0.470$, respectively.

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 content they access. 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 content (measured in terms of how frequently that content is accessed over time). In our context, characterizing live content popularity is not meaningful since clients cannot quite revisit the live content. Rather, it is more appropriate to gauge the "interest" of a client in the live content (measured in terms of how frequently that client accesses the various constituent streams of the live content over time). To some extent, client "interest" could be viewed as the popularity of the client as a recipient of live content. This role reversal highlights the "duality" of stored versus live media access when it comes to the active versus passive roles of clients and streams.

IV. Session Layer Characteristics

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

A. 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

[2] 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 smooths out the variability in the arrival process. This is evident by comparing the maximum values of the three plots in Fig. 4.

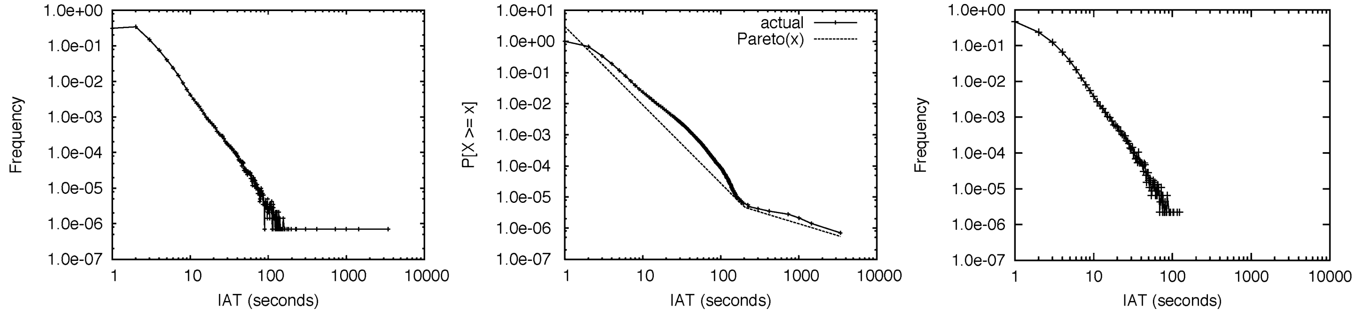

Fig. 6. Client interarrival times: frequency (left) and CCDF (center) marginal distributions of client interarrival times. Frequency marginal distribution of client interarrival times from a piece-wise-stationary Poisson process (right).

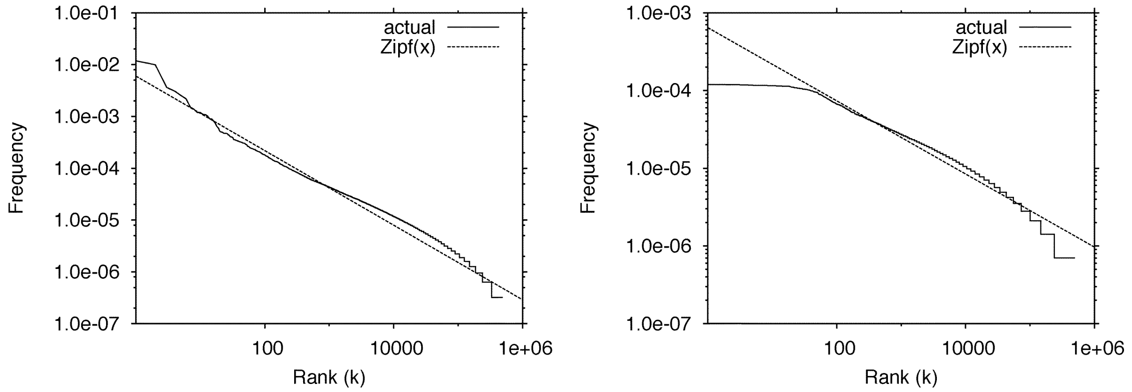

Fig. 7. Client interest profile: relationship between client rank and transfer frequency (left) and session frequency (right).

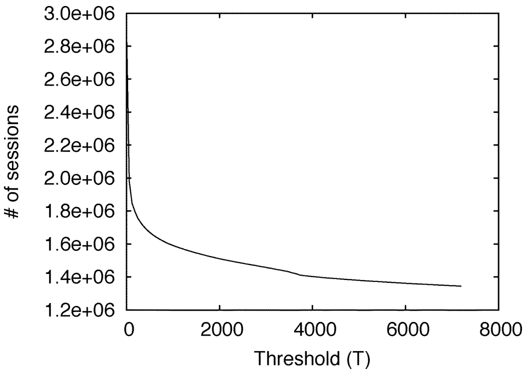

Fig. 8. Relationship between number of sessions and $T_{\rm off}$.

our choice of the session timeout parameter $T_{\rm off}$. Fig. 8 shows the relationship between the number of sessions in the trace and the choice of $T_{\rm off}$. This relationship implies that the number of sessions does not change much beyond $T_{\rm off} > 3600$ seconds (1 hour). For the remainder of this paper, and unless stated otherwise, we use $T_{\rm off} = 3600$ seconds.

B. Session ON Time

To characterize the period of time during which a session is active, we use a time series $l(i)$, which denotes the length of the ith session in the trace. Clearly, $l(i)$ is the ON time for session i. Fig. 9 shows the frequency marginal distribution of $l(i)$ for all sessions identified in the trace. The distribution was fitted to a Lognormal distribution with parameters $\mu = 5.19$ and $\sigma = 1.44$ (also shown in the figure).

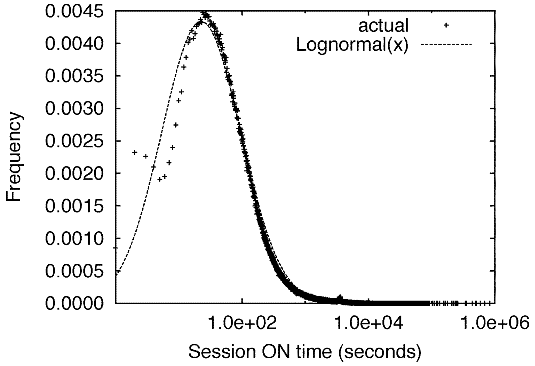

Fig. 9. Distribution of session ON times.

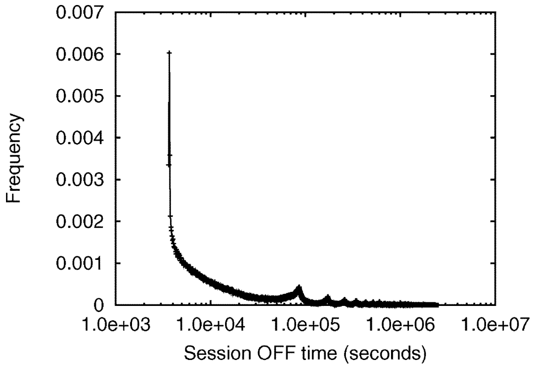

Fig. 10. Distribution of session OFF times.

Fig. 9 indicates that session ON times are highly variable. To determine whether this variability is fundamental to the nature

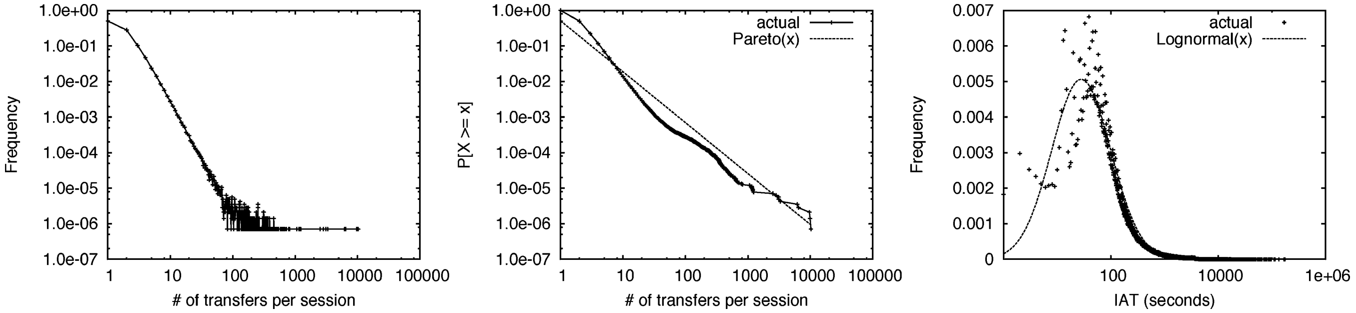

Fig. 11. Frequency (left) and CCDF (center) marginal distributions of number of transfers per session and frequency marginal distribution of session transfer interarrivals (right).

of client interactions with live content or whether it is symptomatic of nonstationarity due to temporal correlation (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. We identified 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 diurnal behaviors (as was the case with number of active clients), but rather it is a fundamental property of the interaction between users and live content. There was further evidence of this when we compared the session lengths at "special times" with those at other times. For example, sessions starting on Sunday evenings seem to be generally shorter indicating that users are checking in for a shorter time on average to check out developments related to which contestant will be eliminated. Sessions started at the start of a workday seem to last longer on average, perhaps an indication of a class of user who tune in at the beginning of a day and tune out at the end of the day (thus bumping up the average session length for sessions starting early in the day). Sessions started at other times seem to have a fairly uniform average length.

C. Session OFF Time

In order to characterize the period of time during which a user is inactive, we use a time series $f(i)$, which denotes the session OFF times. We define $f(i)$ as $t(j) - t(i) - l(i)$, where i and j denote two consecutive sessions in the trace that belong to the *same* client. Clearly, $f(i)$ is the session OFF time (or "logoff" time or "inactive OFF" time). Fig. 10 shows the frequency distribution of $f(i)$ for all sessions identified in the trace.

Fig. 10 shows that large session OFF times seem to form ripples around specific values, which are around 1 day, 2 days, 3 days, etc. (multiples of 86 400 seconds). This underscores the underlying variability in client interests, namely, those "revisiting" the show daily, or every two days, etc. We found that session OFF times fit well an exponential distribution $\lambda e^{\lambda x}$ with $\lambda = 5.025\mathrm{e} - 06$.

D. Transfers per Session

Session ON times underscore the continued activity of a given user as reflected by a number of transfers within that session. Fig. 11 (left) and (center) shows the frequency and CCDF distributions 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 Pareto distribution ab^a/x^{a+1} with parameters $a = 1.43$ and $b = 0.62$. 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 diurnal characteristics. Thus, we attribute this variability to the nature of client interactions with live content.

E. Interarrivals of Session Transfers

The last variable we characterize at the session layer pertains to the interarrival time between transfers within the same session. Large interarrivals would correspond to a fairly passive user behavior, whereas small interarrivals would correspond to users constantly switching from one stream to another (akin to "channel surfing"). Fig. 11 (right) shows the frequency distribution of transfer interarrivals within a single session, which we fitted to a Lognormal distribution with parameters $\mu = 4.93$ and $\sigma = 1.26$ $(\mathrm{mean} = \exp(\mu + 0.5\sigma^2) = 306$ seconds). Our characterization suggests that the interarrival times between transfers within the same session are rather large (average is more than 5 minutes). This can be explained by noting the fact that there are only two streams to choose from; and thus, a flip-flop behavior between the two streams is not likely (and if a client is really interested, he/she could simply have both streams concurrently delivered). Clearly this may well be different if users had more choices—channel surfing is more likely as the number of channels increases and the possibility of viewing all interesting channels concurrently becomes infeasible.

V. 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.

A. 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. Fig. 12 (left) shows the cumulative distribution of the number of concurrent transfers over the entire duration of the trace. We

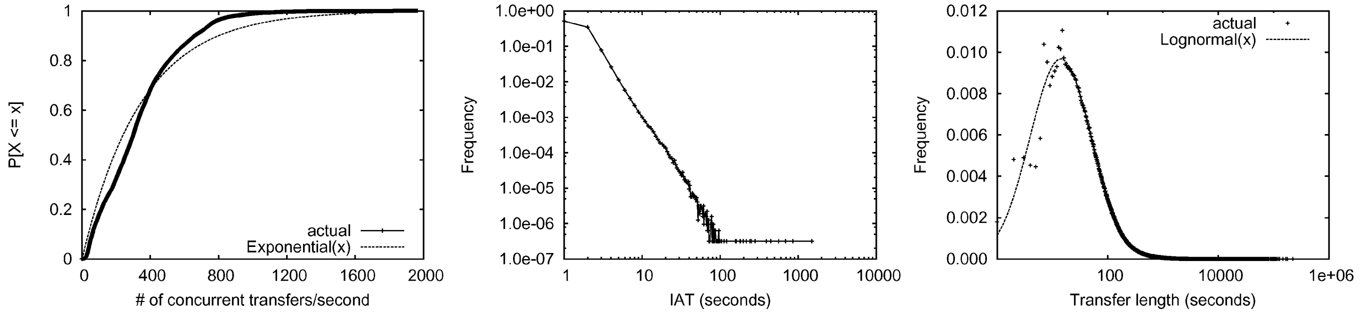

Fig. 12. Transfer layer characteristics: cumulative marginal distribution of concurrent transfers over all sessions (left), frequency marginal distribution of transfer interarrival times (center), frequency marginal distribution of transfer lengths (right).

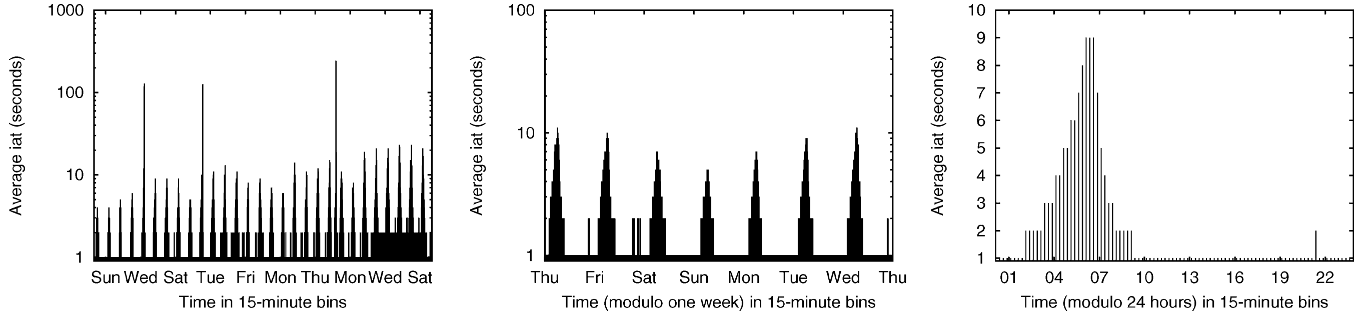

Fig. 13. Temporal behavior of transfer interarrival times: over entire trace (left), daily (center), and hourly (right).

also identified the mean number of active transfers over the entire trace (the figure was not showed in the paper). Not surprisingly, these distributions are fairly similar to those we observed for the number of concurrent clients over time (Figs. 3 and 4).

B. Transfer Length and Client Stickiness

We now turn our attention to the length of time of individual transfers. 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 streams. Let $l(j)$ denote the length (in seconds[3]) of the jth transfer in the trace. Fig. 12 (right) shows the frequency marginal distribution for $l(j)$, which fits a Lognormal distribution with parameters $\mu = 4.29$ and $\sigma = 1.28$ (mean $= \exp(\mu + 0.5\sigma^2) = 166$ seconds).

The size distribution of individual Internet (unicast) transfers has been studied extensively 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], [27] 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.

For live media content workloads, the long tail of the transfer length distribution 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 stream being transmitted. Recall that for live media, the transfer length is 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, nonstreaming media workloads, but rather to the willingness of a client to "stick" to a transfer. It is important to note that for stored streaming content, both stream size and client interactivity play a role in the length of transfers.

C. Transfer Interarrivals

We characterize the transfer interarrival times using a time series $a(j)$ that denotes the interarrival time of the jth and $(j+1)$th transfers. We define $a(j)$ as $t(j+1) - t(j)$, where $t(j)$ denote the starting time of the jth transfer in the trace. Fig. 12 (center) shows the frequency marginal distribution of $a(j)$, which suggests a heavy-tailed nature for this characteristic, which we fitted to a Pareto distribution with parameters $a = 1.263$ and $b = 0.008$.

Like client arrivals, the request arrival process is clearly not stationary. In Fig. 13, we show the periodic nature of that process by plotting the average request interarrival time over the entire 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

[3] Given the real-time nature of live transmission, we use seconds to characterize transfer lengths. Converting the characteristics to "bytes" would be a function of the transfer rate, which we characterize later.

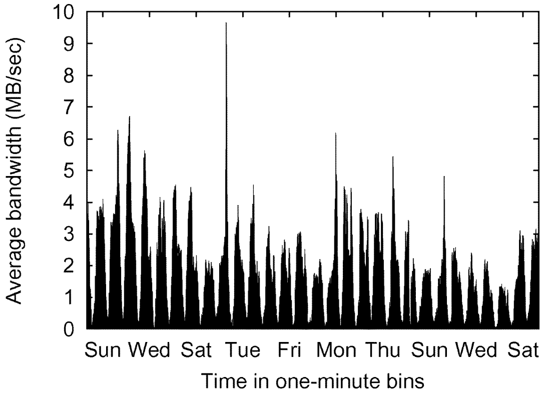

Fig. 14. Aggregate bandwidth (one-minute averages).

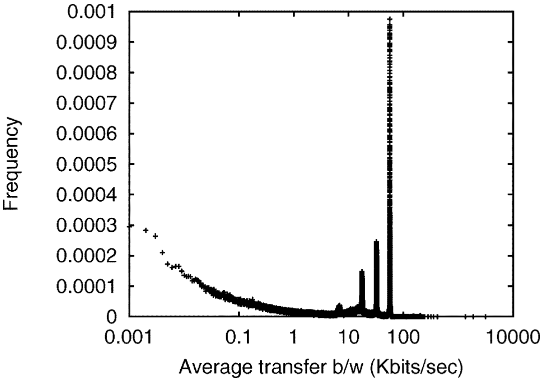

Fig. 15. Frequency distributions of transfer bandwidth.

have lower average interarrivals than weekdays), Fig. 13 indicates that diurnal behaviors are the main source of variability (with 2am to 9am showing considerably longer interarrivals).

D. Transfer Bandwidth

Fig. 14 shows the aggregate server bandwidth. Each point in that plot corresponds to the average bandwidth consumed over a one-minute interval. The figure shows significant *periodic* variability over four orders of magnitude, with peak (one-minute average) values approaching 80 Mb/s.

Fig. 15 shows the marginal distribution of the aggregate bandwidth of Fig. 14, i.e., the distribution of bandwidth experienced by individual 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 the resolution of the encoding chosen by the client (presumably to match various modem speeds of DSL, cable modems, etc.) The value of the bandwidth at these spikes was measured to be 58.6 kb/s, 32.5 kb/s, 17.6 kb/s, and 6.87 kb/s. The second is exemplified by the more uniform values of bandwidth on the left-hand-side of the distribution as well as between the aforementioned spikes, which correspond to *congestion-bound* bandwidth values, resulting from limited network resources and hence a degradation in quality from the prescribed encoding rates. As discussed in Section II, overloaded server (CPU/network) resources are not culprits, making us believe that network congestion was primarily to blame for this degradation. We estimate that around 15% of all transfers were congestion-bound.

VI. Representativeness of Findings

The previous sections summarized our findings with regard to the characterization of a single (albeit substantial) live streaming workload. We contrasted the discovered characteristics to those established in prior work for stored streaming delivery workloads. In this section we elaborate on the representativeness of the trace we considered by providing cursory comparisons with other stored and live streaming media workloads we analyzed.

A. Live Versus Stored Content

Earlier in this paper, we have made several references to the impact of “live” content on the characteristics we observed in the workload. In particular, we hypothesized that in live streaming workloads, temporal (diurnal) characteristics are mostly due to the content as opposed to client behaviors, and thus are not likely to be “smoothed out” by the existence of multiple client population time-zones. To verify this hypothesis, we have analyzed the access patterns to *pre-recorded stored* streams available to the *same* client population at the *same* service provider. Our findings based on over 700 000 requests to 27 821 distinct pre-recorded streams over an 18-day period are shown in Fig. 16. Comparing the results in these figures with those we presented earlier for the live reality show trace in Fig. 4, we observe stronger diurnal patterns for the live content when compared to the stored content. Specifically, comparing Fig. 4 (right) and Fig. 16 (right), the ratio of the maximum to minimum number of active clients is around 16 for live content and only 9 for stored content.

B. Across Multiple Live Media Workloads

A natural question to ask is whether our findings are unique to the workload at hand, or they are representative of live streaming content delivery. To answer this question requires a systematic characterization of a wide range of live streaming workloads to allow for the identification of invariants.

As a step in this direction, we obtained and analyzed the server logs of a second live streaming media content delivered over the Internet. This second live streaming server is for a “news and sports” radio station, which broadcasts live soccer games as well as live (entertainment/sports/travel/weather) news and interviews with soccer players. This second workload consisted of 28,558 requests from 12 867 distinct clients, over a two-week period from mid January 2002 to mid February 2002. Clearly, the nature of the content served by this live “news and sports” streaming server is radically different from that of the live “reality show” streaming server we considered earlier.

We conducted a hierarchical characterization of that second workload and our findings were surprisingly similar (modulo parametrization) to those discussed earlier at all three layers of our hierarchy. Table II compares the various characteristics for the two workloads at the client, session, and transfer layers.

One clear difference between the characteristics of the two workloads concerns the interarrival times (of clients, sessions, and transfers). For instance, the interarrival time of clients was found to follow a Pareto distribution in the reality show workload (see Fig. 6), but was found to follow a Lognormal distribution in the news and sports workload [see Fig. 17 (left)]. We

TABLE II
SUMMARY OF THE DISTRIBUTIONAL CHARACTERISTICS OF THE "REALITY SHOW" AND "NEWS AND SPORTS" LIVE STREAMS

	Live reality show		Live news & sports	
Workload Variable	Distribution	Parameter(s) †	Distribution	Parameter(s) †
Client Interest (transfers)	Zipf	α = 0.719, β = 0.006	Zipf	α = 0.609, β = 0.011
Client Interest (sessions)	Zipf	α = 0.470, β = 0.001	Zipf	α = 0.504, β = 0.005
Number of Active Clients	Exponential	λ = 0.0019	Exponential	λ = 0.0463
Client Interarrival Times	Pareto	a = 2.520, b = 1.550	Lognormal	μ=3.59, σ=1.52
Number of Transfers per Session	Pareto	a = 1.43, b = 0.62	Pareto	a = 1.68, b = 0.39
Session ON Time	Lognormal	μ=5.19, σ=1.44	Lognormal	μ=5.74, σ=2.01
Session OFF Time	Exponential	λ = 5.025e-06	Exponential	λ = 6.008e-06
Session Transfer Interarrival Times	Lognormal	μ=4.93, σ=1.26	Exponential	λ = 0.00114
Number of Concurrent Transfers	Exponential	λ = 0.0029	Exponential	λ = 0.0496
Transfer Length	Lognormal	μ=4.29, σ=1.28	Lognormal	μ=5.08, σ=2.03
Transfer Interarrival Times	Pareto	a = 1.263, b = 0.008	Lognormal	μ=3.09, σ=1.43

† The exponential distribution is of the form $\lambda e^{-\lambda x}$. The Zipf distribution is of the form $\beta x^{-\alpha}$. The Pareto distribution is of the form $ab^a x^{-a-1}$. The Lognormal distribution is of the form $\frac{1}{\sqrt{2\pi}\sigma x} exp\{-(\log(x) - \mu)^2/2\sigma^2\}$.

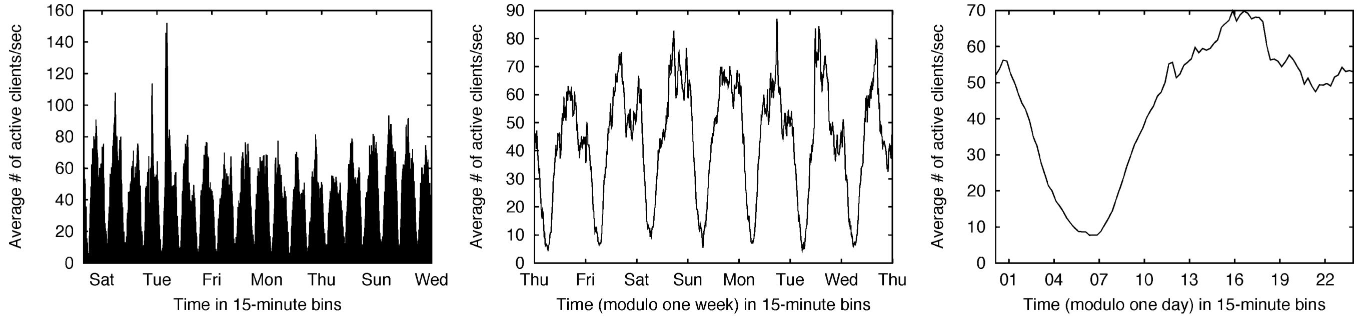

Fig. 16. Stored: temporal behavior of number of active clients: over entire trace (left), daily (center), and hourly (right).

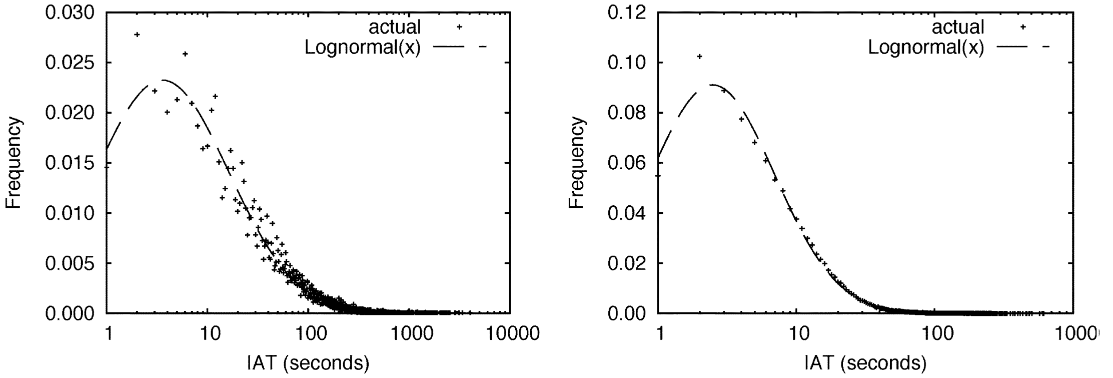

Fig. 17. Frequency marginal distribution of client interarrival times: live radio (left), stored (right).

attribute this difference to the nature of interactions between clients and live streams in the workloads. Specifically, one may argue that a "news and sports" workload features less live content given the periodic/repetitive nature of news programs, as opposed to the spontaneity of a reality show or a soccer game. Indeed, the client interarrival time for the news and sports radio workload resembles that for the pre-recorded stored streams workload, both of which were fitted best to a Lognormal distribution as shown in Fig. 17.

VII. SYNTHESIS OF LIVE MEDIA WORKLOADS

As we discussed earlier, live media workload characterization is crucial to the generation of synthetic (and parameterizable) workloads. In this section, we describe how the results of our hierarchical characterization are used to extend GISMO [21] to generate live media workloads.

A. A Model for Synthetic 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.*

TABLE III
SUMMARY OF THE VARIABLES RETAINED FOR THE SYNTHESIS OF LIVE STREAMING MEDIA WORKLOADS IN GISMO

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 6
Client Interest Profile	Zipf	α = 0.470, β = 0.001	Figure 7
Transfers per Session	Pareto	a = 1.43, b = 0.62	Figure 11 (left)
Interarrival of Session Transfers	Lognormal	μ = 4.93, σ = 1.26	Figure 11 (right)
Transfer Length	Lognormal	μ = 4.29, σ = 1.28	Figure 12 (right)

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.[4]

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 nonstationary Poisson process whose mean is keyed to the periodic behavior of Fig. 4. To determine *which* client should be associated with a given arrival, we use the client interest profile of Fig. 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 Fig. 11 (left).

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, 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 Fig. 11 (right). The length of each transfer is determined using the distribution of transfer lengths shown in Fig. 12 (right).

Table III summarizes the subset of variables we retained in our generative model, as well as the specific distribution parameters suggested by our characterization of the workload at hand. It is important to note that—as we surmised at the outset and as we established by contrasting the reality show and the news and sports workloads—some of the characteristics of live media workloads are likely to depend on the nature of the applications at hand. For example, the periodicity observed in a reality show workload is likely to be different from that observed in live feeds for a soccer game. That said, 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 [21]. For example, in Table III the interarrival of session transfers would have to be changed from Lognormal to exponential for the live news and sports application characterized in Table II.

B. 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 prerecorded 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 "dummy" streams (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 streams. 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 nonstationary of arrivals in GISMO. We do so by allowing the parameters of the arrival processes to be programmable, e.g., by using a user-supplied diurnal patterns.

From our analysis of clients interest in 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 a stream 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 streams to reflect stream popularity and/or client interest profiles.

VIII. RELATED WORK

Workload characterization is fundamental to the synthesis of realistic workloads. Many studies focused on the characterization and generation of nonstreaming (such as HTTP) workloads (e.g., [4]–[9], [14], [15], [19], [30], [31]). These studies have improved our understanding of the nature of access patterns involving stored, nonstreamed content (e.g., documents). Some of the important findings of these studies include the characterization of Zipf-like document popularity distribution, heavy-

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

tailed object and request size distributions, and reference locality properties. A discussion of the various characteristics of workloads involving nonstreamed content 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 [2], [3], [11], [20], [29], have characterized workloads of pre-recorded media object access primarily from media servers for educational purposes. We summarize these efforts below.

Padhye and Kurose [29] studied the patterns of user interactions with a media server. 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). The distributions of both ON period and OFF period appeared to be heavy-tailed—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]. They found there was strong temporal locality of reference. Accesses exhibited geographical locality, i.e., a small number of local machines accounted for most 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.

Chesire *et al.* [11] analyzed a 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.

Almeida *et 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. 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 revealed a high degree of user interactivity in the workload, which implied that the effectiveness of multicast delivery is limited [22].

Streaming Traffic Characterization: Several studies [10], [23], [24], [35] have focused on low-level dynamics of streaming access, such as packet loss and delay, network transport protocols.

Mena and Heidemann [24] 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 packet sizes and 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 [23] analyzed performance metrics such as 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 [35] analyzed RealVideo traffic from several Internet servers to geographically diverse users. They found that typical RealVideos achieve a reasonably high quality. Video performance is most influenced by the bandwidth of the end-user connection to the Internet, but high-bandwidth Internet connections push the performance bottleneck closer to the servers.

Merwe, Sen and Kalmanek [33] presented results from a cursory characterization of two types of streaming workloads on the Internet: on-demand streaming of pre-recorded content and live broadcasting. Their study revealed that requests for high-bandwidth encodings are more prevalent than low bandwidth ones (with a two-to-one margin), that the traffic resulting from high-bandwidth encodings dominates in terms of byte traffic, that Microsoft Windows Media is the dominant media type, and that TCP is the transport of choice (with more than a two-to-one margin over UDP). Also, their study revealed that a small percentage of routing prefixes accounted for most of the traffic demand, which suggests that substantial bandwidth efficiency can be realized using replication and CDN. While their work highlighted some differences between live and stored media workloads, it did not construct a model or suggest distributional characteristics for live streaming workloads as we have done.

IX. Summary and Conclusions

In this paper we have presented a thorough characterization of what we believe to be the first significant *live* Internet streaming media workload in the scientific literature. We adopted a hierarchical approach at three layers, corresponding 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 Pareto 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 stickiness, 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 stored object workloads, whether streamed (e.g., pre-recorded media objects) or not (e.g., files). The difference stems from the role reversal of objects and clients in live versus stored content delivery. Accesses to stored streaming objects are *user driven*, whereas accesses to live streaming objects are *content driven*. This observation, together with the results of our 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.

ACKNOWLEDGMENT

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

REFERENCES

[1] S. Acharya and B. Smith, "An experiment to characterize videos stored on the Web," in *Proc. ACM/SPIE Multimedia Comput. Netw. (MMCN)*, Jan. 1998, pp. 166–178.

[2] S. Acharya, B. Smith, and P. Parns, "Characterizing user access to video on the World Wide Web," in *Proc. ACM/SPIE Multimedia Comput. Netw. (MMCN)*, Jan. 2000, pp. 130–141.

[3] J. Almeida, J. Krueger, D. Eager, and M. Vernon, "Analysis of educational media server workloads," in *Proc. NOSSDAV*, Jun. 2001, pp. 21–30.

[4] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, "Characterizing reference locality in the WWW," in *Proc. Int. Conf. Parallel Distrib. Inf. Syst. (PDIS)*, Dec. 1996, pp. 92–107.

[5] M. Arlitt and C. Williamson, "Internet Web servers: workload characterization and performance implications," *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 631–645, Oct. 1997.

[6] G. Banga and P. Druschel, "Measuring the capacity of a Web server," in *Proc. USENIX Symp. Internet Technol. Syst. (USITS)*, Dec. 1997, pp. 61–71.

[7] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in Web client access patterns: characteristics and caching implications," *World Wide Web*, vol. 2, no. 1, pp. 15–28, 1999.

[8] P. Barford and M. Crovella, "Generating representative Web workloads for network and server performance evaluation," in *Proc. ACM SIGMETRICS*, Jun. 1998, pp. 151–160.

[9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proc. IEEE INFOCOM*, Apr. 1999, pp. 126–134.

[10] K. C. Lan and J. Heidemann, "Multi-scale validation of structural models of audio traffic," USC Information Sci. Inst., Marina del Rey, CA, Tech. Rep. ISI-TR-544, 2001.

[11] M. Chesire, A. Wolman, G. Voelker, and H. Levy, "Measurement and analysis of a streaming workload," in *Proc. USENIX Symp. Internet Technol. Syst. (USITS)*, Mar. 2001, pp. 1–12.

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

[13] Windows Media Services 4.1, Microsoft Windows Media. [Online]. Available: http://www.microsoft.com/windows/windowsmedia/default.mspx

[14] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[15] C. Cunha, A. Bestavros, and M. Crovella, "Characteristics of WWW client-based traces," Comput. Sci. Dept., Boston Univ., Boston, MA, Tech. Rep. BU-CS-95-010, Apr. 1995.

[16] A. B. Downey, "The structural cause of file size distributions," in *Proc. MASCOTS*, Aug. 2001, pp. 361–370.

[17] IPAS software for IP address to autonomous system number convertion, National Lab. Applied Network Research's Measurement and Operations Analysis Team (MOAT). [Online]. Available: http://moat.nlanr.net/Software/IPAS/

[18] IPV4 BGP geopolitical analysis, Cooperative Association for Internet Data Analysis (CAIDA). [Online]. Available: http://www.caida.org/analysis/geopolitical/bgp2country/

[19] S. D. Gribble and E. A. Brewer, "System design issues for Internet middleware services: deductions from a large client trace," in *Proc. USENIX Symp. Internet Technol. Syst. (USITS)*, Dec. 1997, pp. 207–218.

[20] N. Harel, V. Vellanki, A. Chervenak, G. Abowd, and U. Ramachandran, "Workload of a media-enhanced classroom server," in *Proc. Workshop Workload Characterization*, 1999.

[21] S. Jin and A. Bestavros, "GISMO: generator of streaming media objects and workloads," *Perform. Eval. Rev.*, vol. 29, no. 3, pp. 2–10, 2001.

[22] ——, "Scalability of multicast delivery for nonsequential streaming access," in *Proc. ACM SIGMETRICS*, Jun. 2002, pp. 97–107.

[23] D. Loguinov and H. Radha, "Measurement study of low-bitrate internet video streaming," in *Proc. ACM SIGCOMM Internet Measurement Workshop (IMW)*, Nov. 2001, pp. 281–293.

[24] A. Mena and J. Heidemann, "An empirical study of real audio traffic," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 101–110.

[25] D. A. Menascé and V. A. F. Almeida, *Capacity Planning for Web Services: Metrics, Models, and Methods*. Upper Saddle River, NJ: Prentice-Hall, 2002.

[26] 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 *Proc. ACM Conf. E-Commerce*, Oct. 2000, pp. 56–65.

[27] M. Mitzenmacher. (2002) Dynamic models for file sizes and double Pareto distributions. [Online]. Available: http://citeseer.nj.nec.com/mitzenmacher02dynamic.html

[28] The RealONE SuperPass Subscription Service, Real Networks. [Online]. Available: http://www.real.com

[29] J. Padhye and J. Kurose, "An empirical study of client interactions with a continuous-media courseware server," in *Proc. NOSSDAV*, Jun. 1998.

[30] V. N. Padmanabhan and L. Qiu, "The content and access dynamics of a busy Web site: findings and implications," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 111–123.

[31] V. Paxson, "Wide-area traffic: the failure of Poisson modeling," in *Proc. ACM SIGCOMM*, Aug. 1994, pp. 257–268.

[32] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, Jun. 1995.

[33] J. van der Merwe, S. Sen, and C. Kalmanek, "Streaming video traffic: characterization and network impact," in *Proc. Web Caching Workshop*, Aug. 2002.

[34] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," Comput. Sci. Dept., Boston Univ., Boston, MA, Tech. Rep. BUCS-TR-2002-014, May 2002.

[35] Y. Wang, M. Claypool, and Z. Zuo, "An empirical study of video performance across the Internet," in *Proc. ACM SIGCOMM Internet Measurement Workshop (IMW)*, Nov. 2001, pp. 295–309.

Eveline Veloso received the Masters degree from the Federal University of Minas Gerais, Brazil.

She is interested in large-scale distributed systems, mainly the Internet, and information retrieval.

Virgílio Almeida received the Ph.D. degree from Vanderbilt University, Nashville, TN.

He is a Professor in the Computer Science Department, Federal University of Minas Gerais, Brazil. His research interests include performance evaluation and modeling of large-scale distributed systems. He has held visiting professor positions at Boston University, Boston, MA, and the Polytechnic University of Catalunya, Barcelona, Spain.

Wagner Meira, Jr. received the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 1997.

He is currently an Associate Professor in the Computer Science Department, Federal University of Minas Gerais, Brazil. His research interests are large-scale parallel and distributed systems, and data mining algorithms and their applications.

Azer Bestavros (M'87) received the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA.

He is currently Professor and Chairman of Computer Science at Boston University, Boston, MA. His research interests are in networking and real-time systems. His seminal works include his generalization of the classical RMS, his pioneering of the push model for Internet content distribution, and his characterization of Web traffic self-similarity and reference locality. He has served as chair, officer, or PC member of major conferences in real-time and networking systems.

Dr. Bestavros has received the ACM and the IEEE Excellence Awards for services rendered to the Computer Science community, including his organization and PC chairmanship of a number of IEEE and ACM technical meetings, and his maintenance of the archives of the IEEE Computer Science Technical Committee on Real-Time Systems. He has been a member of the Association for Computing Machinery (ACM) since 1987.

Shudong Jin (M'00) received the B.S. and M.S. degrees in computer science from Huazhong University of Science and Technology, China, in 1991 and 1994, respectively. He received the Ph.D. degree in computer science from Boston University, Boston, MA, in 2003.

He is currently an Assistant Professor in computer science at Case Western Reserve University, Cleveland, OH. His research interests include network protocols and algorithms, network modeling and performance evaluation, multimedia streaming, and pervasive computing.

Dr. Jin has been a member of the Association for Computing Machinery (ACM) since 2001.