Understanding Couch Potatoes: Measurement and Modeling of Interactive Usage of IPTV at large scale

Vijay Gopalakrishnan, Rittwik Jana, K. K. Ramakrishnan,
Deborah F. Swayne, Vinay A. Vaishampayan
AT&T Labs Research, 180 Park Ave, Florham Park, NJ, 07932
{gvijay,rjana,kkrama,dfs,vinay}@research.att.com

ABSTRACT

We investigate how consumers view content using Video on Demand (VoD) in the context of an IP-based video distribution environment. Users today can use interactive stream control functions such as skip, replay, fast-forward, pause, and rewind to control their viewing. The use of these functions can place additional demands on the distribution infrastructure (servers, network, and set top boxes) and can be challenging to manage with a large subscriber base. A model of user interaction provides insight into the impact of stream control on server and bandwidth requirements, client responsiveness, etc.

We capture the activity users in a natural setting, viewing video at home. We first develop a model for the arrival process of requests for content. We then develop two stream control models that accurately capture user interaction. We show that stream control events can be characterized by a finite state machine and a sojourn time model, parametrized for major periods of usage (weekend and weekday). Our semi-Markov (SM) model for the sojourn time in each stream control state uses a novel technique based on a polynomial fit to the logarithm of the Inverse CDF. A second constrained model (CM) uses a stick-breaking approach familiar in machine learning to model the individual state sojourn time distributions. The SM model seeks to preserve the sojourn time distribution for each state while the CM model puts a greater emphasis on preserving the overall session duration distribution. Using traces across a period of 2 years from a large-scale operational IPTV environment, we validate the proposed model and show that we are able to faithfully predict the workload presented to a video server. We also provide a synthetic trace developed from the model enabling researchers to also study other problems of interest. We also use the techniques to model consumer viewing of video content recorded on their personal Digital Video Recorder (DVR).

IMC'11, November 2–4, 2011, Berlin, Germany.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

General Terms

Measurement, Performance

Keywords

Video-on-Demand, IPTV, Data Mining

1. INTRODUCTION

Viewers are increasingly watching stored video, delivered over IP networks, in preference to broadcast television. The new modes of viewing content offer greater interactive control; at the same time, they place additional demands on the distribution infrastructure (servers, network, and set top boxes), which can be challenging to manage with a large subscriber base. In addition to video quality, a consumer's viewing experience is determined by the system's responsiveness to stream control operations (e.g., pause, fast-forward, rewind). A poorly engineered and unresponsive system leads to confusion for the user about whether a stream control event was registered by the system, generates unnecessary additional requests, and ultimately leads to decreased user satisfaction. A provider's infrastructure must be able to provide a minimum level of responsiveness despite the demands that user interactions place on the set-top box, the network, and the server.

Knowing how consumers use stream control features allows the provider to optimize the delivery system and make it responsive. This includes provisioning the system with the requisite number of servers, network bandwidth, and set-top box processing, identifying the right delivery mechanisms (unicast, multicast or P2P), choosing the right method for serving functions such as fast-forward and rewind, and fine-tuning timers and buffering. In order to correctly engineer the servers and network to support such interaction, providers need to have a good understanding of users' interactive behavior.

This paper provides a comprehensive model of a large population of consumers downloading and interactively viewing Video-on-Demand (VoD) content. Our overall model consists of two main components: an arrival process model and a stream control usage model. We model the arrival process for VoD requests, with the objective of capturing the distinctive diurnal pattern and having a traffic intensity that can be scaled as a function of the number of users provisioned in a VoD system. Our model for stream control usage enables us to get a better understanding of the usage of these functions than ever before.

Previous studies of user interactivity have either been based on a small set of users under laboratory conditions [3] or with P2P users [22]. With the former, users are not in their natural environment and are aware of being monitored. With the latter, responsiveness can be dramatically affected by peer bandwidth and latency and users tend to adapt their behavior to the system's responsiveness. As a result, inference based on these approaches can lead to biased results.

In contrast, to develop our model, we use comprehensive data collected over a long period of time from a nationally deployed IPTV service. Our data includes users requesting VoD content and all the stream control requests generated. The data is obtained from a large population of users viewing television programs and videos over a long period in a natural setting – in their own homes. Our data spans a period of two years, capturing two representative weeks from each of the four different seasons. We used this large data set to understand the variability in VoD viewing that may be influenced by seasonal patterns in television programming and human behavior. We also use this data to characterize the user's interaction with their DVR for viewing recorded content.

To characterize the arrival process of user requests for VoD content, we have examined a total of 120 days of data collected from nine metropolitan areas, all of which has been used for characterizing the arrival process. The average number of set-top boxes provisioned was approximately 3 million over this period.

We characterize a user's stream control interactions for both VoD and DVR based on analyzing detailed traces of *all* interactions generated by a more limited population of about 300K users over a period of 9 days from 10 metropolitan areas (to capture both weekend and weekday behavior), while viewing either videos from the provider's VoD library or recordings on their in-home DVR. Even though the stream control modeling work reported here is based on traces covering the 9 day period, we have verified the validity of our model structure using the traces from the longer 2-week periods spanning the 2 year interval.

As a preview of one of the main results in the paper, we show the state transitions that model the stream control operations performed by a typical user (our typical couch potato) in Fig. 1. As an example of the information the figure conveys, note that while viewing VoD (and thus in the PLAY state), the most common action a viewer initiates is FASTFORWARD (FF), followed by PAUSE and REWIND. We provide the details of the model, its derivation and the sojourn times in each state later in this paper. As will be evident, our typical couch potato is an active participant in interactive viewing of on-demand content. This has implications on the system design, especially as on-demand viewing grows.

The contributions of this paper are as follows.

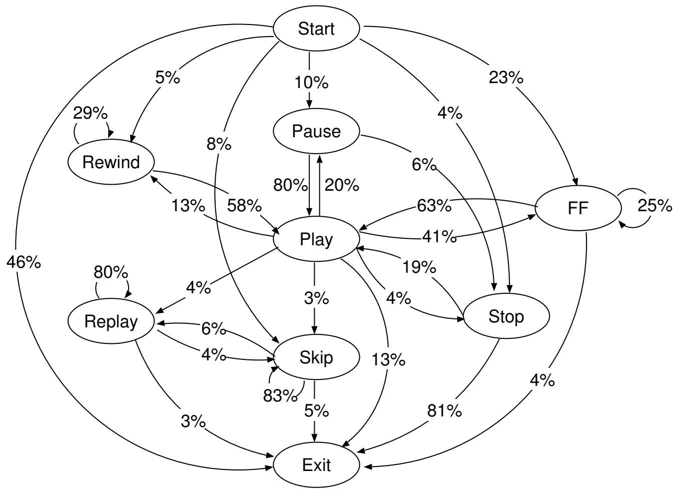

Figure 1: The weekend version of our synthetic couch potato for VoD.

1. *Arrival Process model:* We develop a VoD request arrival process model based on data from two years. The arrival process is parametrized by a set of K Fast Fourier Transform coefficients which can be inverted to generate arrival counts by the second for a given weekday or weekend. This enables us to generate an aggregate stream control arrival process at a VoD server. Each arrival of a request translates to a session that contains multiple stream control events.

2. *Stream control model:* We develop two stream control models. Each model uses a common finite state machine (FSM) in order to model the sequence of stream control events in a given session. A state in the FSM is a particular type of stream control event (e.g. REWIND or PLAY). The models differ in the methods by which the state sojourn times are generated. In the first model, state sojourn times are assumed to be independent and are fitted using a polynomial function of the CDF on a logarithmic scale. This sojourn-time model is referred to as the independent sojourn time model (IS). Thus, the first model is a conventional semi-Markov model (SM = FSM + IS) and will be referred to as such. In our second model, we constrain the state sojourn times in order to preserve the aggregate session duration distribution. The approach follows the well-known stick-breaking (SB) paradigm, as it is commonly referred to in the statistical literature [14]. We call the second model the *constrained model* (CM = FSM + SB). We use these techniques to also model stream control events for a user viewing recorded content on the in-home DVR.

3. *Validation:* We provide extensive validation of our model by using a simulator to compare the server load and interruptions to the user viewing the content (because of the under-run of the client playout buffer) from a purely synthetically generated workload with that of a real trace, and showing a good match on other significant statistics as well.

4. *Public data:* We provide synthetic traces of stream control events that researchers can use to validate other algorithms [9].

The main body of the paper is devoted to describing the modeling process. Although we recognize that it would be ideal to provide a real trace (or traces) to readers so as to enable them to arrive at their own models as well as use it to study other problems of interest, we are limited by privacy agreements and legal constraints. However, we provide the 'next best' thing, a synthetic trace for a weekday and weekend day that we believe would serve the same purpose. In addition, we believe the complete model parameters that we provide in [9] will enable readers to generate their own synthetic trace for a given population of users.

2. RELATED WORK

Modeling interactive behavior is important in designing any networked system which offers interactive usage, such as Web browser navigation and IPTV systems, and a variety of data mining techniques have been used to analyze such data (e.g., click stream modeling [1, 17]). User behavior is typically modeled using Markov processes [3]. Such modeling work often focuses on predicting individual user behavior in order to provide personalized services, but our interest is in global behavior so that we can characterize the workload of a centralized VoD server.

Branch et al. [3] report on the statistics of user behavior for a VoD based system of 63 viewing sessions conducted in a controlled setting. They report that lognormal sojourn time distributions are good approximations for all the states of the Markov state machine. In addition to the potential limitations of characterizing user behavior in a controlled setting, we find that a lognormal distribution does not adequately capture the complete statistics (mean, variance, third moment and median). Furthermore, this study also lacked stream control functions for discontinuous viewing, such as "skip" and "replay".

Mongy et al. [18] investigate user behavior by clustering video sessions to measure similarity between session groups. From the standpoint of clustering user behavior, we identified that weekday and weekend sessions are sufficiently different in their VoD server requirements that they must be modeled separately.

There have also been several studies on modeling access patterns for Web-based streaming video systems. Cha et al. [4] have investigated how users access videos in YouTube. Guo et al. [11] compared access patterns for live streaming, Web, P2P and VoD. They do not specifically model the individual stream control operations. We believe that stream control operations have a significant impact on the server load and it is therefore important to understand and model these operations.

Several other studies have sought to characterize P2P-TV and P2P-VoD traffic on the Internet [13]. [15] and [21] describe designs for peer-assisted VoD solutions that offer capabilities such as fast forward and seek to random play points. Our work on a system to cooperatively use peer assists and multicast [8] also seeks to provide a complete suite of stream control operations for VoD usage in a large-scale IPTV distribution environment. Such designs could benefit from a comprehensive model for stream control operations that a typical VoD user would generate. We believe our proposed model can guide the design of such P2P-VoD systems.

More recently, Qiu et al. [20] and Cha et al. [5] have investigated modeling the user activities for an IPTV system but have primarily focused on LiveTV sessions. Their model captured events related to channel switching and STB power on/off events. A synthetic workload generator is proposed to estimate the load on a server due to channel change events, taking as input the arrival process of an aggregate set of

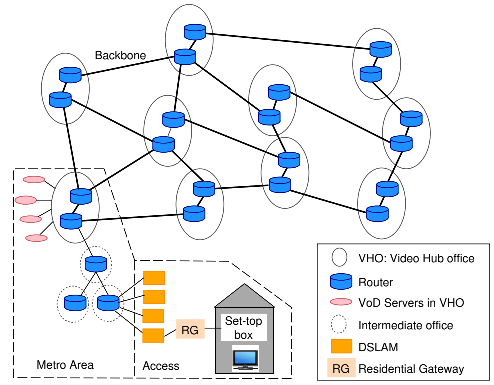

Figure 2: Architecture of a typical IPTV network

users. We believe that for VoD it is equally important to model the individual stream control events as these have a direct impact on the VoD server. Yin et al. [23] present a data characterization study of a live Internet VoD system for the Beijing Olympics. They study how presentation models (e.g., instant messaging and advertising) impact user behavior. To the best of our knowledge, ours is the first study of user stream control operations based on a commercially deployed IPTV system at national scale.

3. VOD USAGE

In this section, we describe a typical IPTV network, describe the data we use in this paper, and finally present the characteristics of stream control events.

3.1 System Architecture

Fig. 2 shows the typical architecture used for IPTV delivery. These networks consist of a set of interconnected Video Hub Offices (VHO), each of which serves one or more metro areas. All the customers in a metro area connect to a Video Hub Office (VHO) and receive content from the VoD servers in that VHO using unicast [8]. When a customer watching a video performs a stream control event that can not be satisfied using the local buffer, a request is transferred to the server to execute the appropriate action. DVR use, on the other hand, does not place that sort of demand on the server. The broadcast content recorded on the customer's DVR is delivered using IP-multicast, after which every action performed as part of DVR viewing is local to the customer. In both cases, stream control events, like all other events performed using the remote control, are logged and the logs are uploaded to the VHO. Understanding these events would help in the design of VoD delivery, as well as DVR solutions, as they evolve.

3.2 Data used

Table 1 shows the different snapshots of data that we have used throughout the paper. We have used multiple data sets spanning over two years and covering a large subscriber footprint. DS1 is used for the modeling of the arrival process, DS2 for the stream control finite state machine characterization, and DS3 for characterizing the evolution of video popularity.

We briefly describe the different stream control operations that are available to users; from those operations, we construct the finite state machine (FSM) shown in Fig. 1 and

Data set	Duration (days)	# of VHOs	# of STBs	Time 2009/2010
DS1	120 days	9	3M	Jan/Apr/July/Nov
DS2	9 days	10	300K	Jan
DS3	60 days	10	300K	Jan/Feb

Table 1: Data description

develop it further in subsequent sections. When users select a video, they enter the START state. The video begins playing (equivalent to being in the PLAY state). At that point there is a set of stream control operations that a VoD user has access to: PLAY, FASTFORWARD, REWIND, SKIP, REPLAY, PAUSE, and STOP. Users who do not use any stream control operations in a session transition directly from START to EXIT (users EXIT when they watch the video until the end or explicitly exit before the end).

While most of these operations are self-explanatory, it is important to distinguish between START and PLAY. Users enter the START state when they first start the video; they enter PLAY when they resume playing the video after another stream control operation (e.g., PAUSE or FASTFORWARD). Finally, when they end the session, users enter the EXIT state.

3.3 Stream Control Usage Characterization

Understanding the intensity of stream control operations and their breakdown, as well as the influence of the length of the video, is useful and motivates our modeling work. We briefly describe relevant VoD usage characteristics and refer the reader to a short paper [10] for further details. We have observed that the number of stream control events per session is higher during weekdays than weekends.

For this study, we used the nine days of trace data collected at *ten* VHOs. The data was anonymized to protect the identity of the users. It contained over two million requests for videos from ∼300K users. The data included information about the videos requested by the users, the set of stream control operations performed, and when they were performed. The library contains a large number of videos of different genres and lengths.

Video length (in min)	Weekend Mean	Weekend 95%	Weekday Mean	Weekday 95%
0-10	4.8	13	4.8	13
10-30	8.9	32	8.9	32
30-60	13.2	46	15.2	54
60-120	9.7	39	13.3	54
120-	8.8	33	10.7	41

Table 2: Statistics of the number of events per viewing session by video length.

We first look at the the statistics for stream control events based on the content length (referred to as *video length*). The statistics are provided for five ranges of video lengths in Table 2. We notice that the mean and 95th percentile attain peak values for video lengths in the range 30–60 mins. This indicates that there is a correlation between the number of stream control events and the length of the video, which is expected. We also looked at the the number of stream control events based on video genre, and cost (i.e., paid vs. free videos). We *did not* find significant differences in the

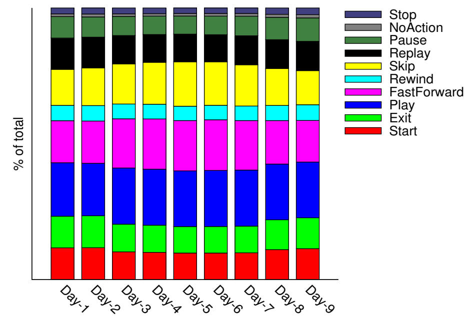

Figure 3: Breakup of VoD stream control events

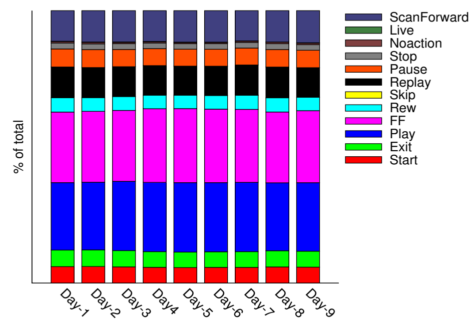

Figure 4: Breakup of DVR stream control events

average number of stream control events in either of the two cases.

Fig. 4 and Fig. 3 show the relative proportion of the various stream control operations. The sum of FASTFORWARD (FF), PLAY, and REPLAY comprise 45% of the total. SKIP is also frequent; FASTFORWARD is very common in DVR viewing (presumably to skip over commercials).

3.4 VoD Content Length and Popularity

We next look at the distribution of the video lengths. Figure 5 shows the video length distributions. We observe that a significant fraction are short videos, and there are about 5 clusters of video lengths.

We also characterize the popularity-rank distribution of the VoD library. Fig. 6 shows the logarithm of video popularity (number of times each video was requested) against the logarithm of the video's rank for one representative day; the dotted reference line has a slope of -1, characteristic of

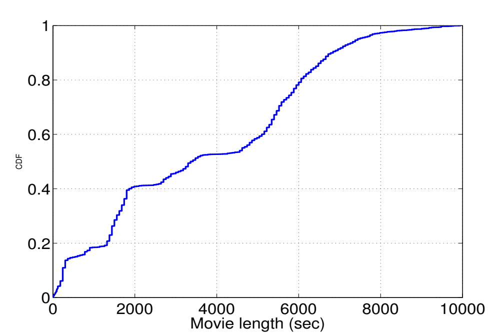

Figure 5: Video length distribution

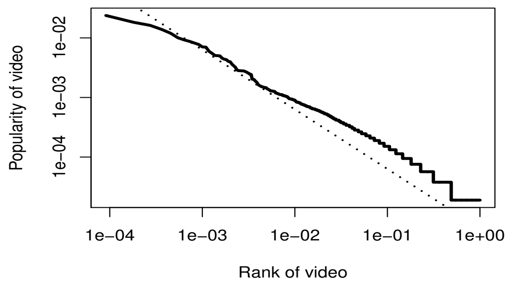

Figure 6: VoD popularity distribution

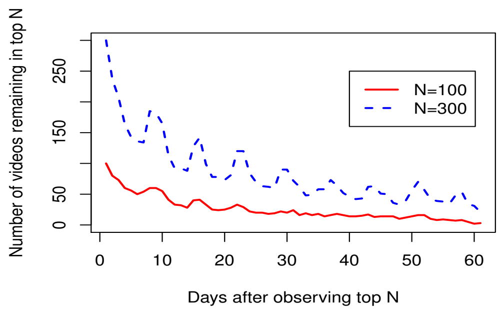

Figure 7: Evolution of VoD popularity

the Zipf distribution, similar to observations made by others (e.g., [4]). This plot is based on all the videos requested by all subscribers over one 24-hour period; we chose a day that falls within our primary 9 day study period. We observe that this popularity distribution is similar to what has been reported in other studies.

In addition to the popularity, we believe it is also useful to understand the evolution over time of the popularity of a particular video. Figure 7 starts on that same day as in the previous plot, and we chose the 300 (or 100) most popular videos as an initial list. For the next 60 days, we counted how many of those popular videos remained among the top 300 (or 100) most popular. We chose a window of 60 days because some videos in the library get aged out and are replaced by new content. We note the rapid drop-off in the popularity of both the top 100 as well as the top 300 videos. Furthermore, there is a periodicity associated with this change, indicating that some videos are more popular during the week and others on weekends. This has implications for several system design issues, such as managing the VoD library, caching, etc.

We utilize the characterization of the video length in our synthetic trace generation. We could potentially use the video popularity in the synthetic trace generation as well.

3.5 Quantifying the Impact of Stream Control

To motivate our study of stream control operations, we examined the impact of these operations using a discrete event simulator that faithfully emulates the interactions between clients and the video server in the IPTV environment. Our primary metric for evaluation is the peak server bandwidth, as it serves up the video requests and processes all the stream control operations of the corresponding video.

Figure 8 shows the peak server bandwidth required over a particular 24-hour day based on the actual trace, first accounting only for the video requests (and ignoring all the stream control operations). Second, for the same trace, all

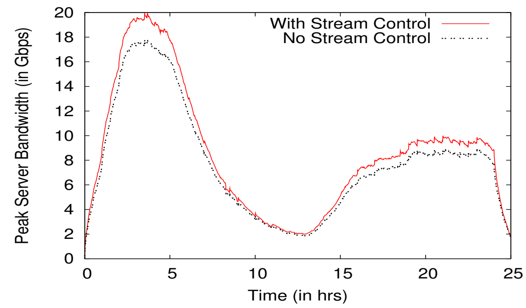

Figure 8: Impact of Stream Control Operations

the stream control operations are also properly accounted for. We observe that server bandwidth increases from about 17 Gbps to about 20 Gbps at the peak when stream control operations are accounted for, a reasonably significant increase. We use this to motivate the need to both model the stream control operations and understand their impact.

4. MODELING ARRIVAL AND STREAM CONTROL PROCESSES

Our objective is to construct a statistical model for the video request arrival process and for a sequence of stream control requests, as seen by a server complex of a large IPTV VOD system. The instantaneous load on a server at a given time, especially for video stream delivery, is determined by the number of concurrent sessions at that time. We define concurrent sessions as all those sessions whose state lies between START and EXIT at a given time. The number of concurrent sessions is determined by the instantaneous arrival rate and instantaneous session duration. Our overall model is thus made up of two components, an arrival (or session-start) process model that mimics the arrivals of new sessions into the system and a stream control model. The stream control model consists of a finite state machine (FSM) that models the detailed stream-control characteristics of a typical customer (the hop sequence from individual state to state), and a model for the sojourn times for each state of the FSM. The arrival process is described in Sec. 4.1, the finite state model in Sec. 4.2 and the sojourn time models in Sec. 4.3. Fig. 9 depicts the overall modeling and validation process. We first generate a sequence of session starts, hereafter referred to as arrivals. Each arrival results in a selection of a video according to the video length distribution described in Sec. 3.4, and spawns a separate state machine which generates a sequence of stream control events. The server sees the aggregate of the arrivals and their stream-control requests generated by each state machine. Our model thus requires that we characterize the arrival process and the state machine with sufficient accuracy.

4.1 Arrival Process Modeling

We first model the arrival process for VoD traffic. Our objective is to retain the shape of the daily profiles (diurnal pattern). The intensity of the profile is however scaled as a function of the number of users provisioned in the system. We noticed that there is a marked increase in the number of arrivals for the weekends compared to the weekdays. For modeling the arrival process, we look at the traces from the

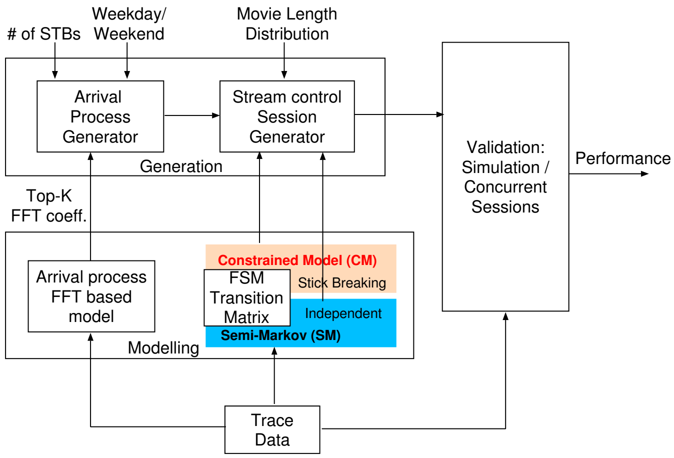

Figure 9: Modeling and validation approach

150 days across 2 years, which includes all 4 seasons of the TV programming cycle.

The arrival process is characterized by the arrival time t_n of the nth customer and the inter-arrival time $\tau_n = t_n - t_{n-1}$ between the nth and $n-1$th customer. If all inter-arrival times are i.i.d random variables with cumulative distribution function (CDF) $F_A(t)$, then $P(\tau_n \leq t) = F_A(t)$. We associate a counting process $N(t), t \geq 0$ to the arrival process $t_n, t \geq 0$ by the equivalence of $N(t) \geq n \iff t_n \leq t$. Thus $N(t)$ is a renewal process with inter-arrival time CDF $F_A(t)$.

Algorithm 1 Arrival process estimation for each day

Require: Input: t_n, D days, $S = \#$ of provisioned STBs; Output: FFT coefficients, F_{dk}.

for $d \in D$ **do**
  /* Binning each second and normalize counts */ $N_d = \text{histogram}(t_n, 86400)/S$
end for
/* Average N_d over all days */ $\bar{N_d} = N_d/D$
/* Perform FFT for average day */ $F_d = \text{FFT}(\bar{N_d})$
/* Save top K FFT coefficients */ $F_{dk} = F_d(topK)$
/* Regenerate arrival counts with inverse FFT */ $\hat{N_d} = \text{IFFT}(F_{dk})$

For each day of the week we provide parameters that can be used to regenerate a representative synthetic arrival process. To model a particular day of the week (e.g. Saturday) we pooled data for 16 Saturdays over a two year period at different times of the year (January, April, July, and December). The following procedure was then used in Algorithm 1. Arrivals were binned by the second and then normalized by the total number of STBs. This was necessary since the number of subscribers was increasing over the two years. We then computed the average daily profile and its Fast Fourier Transform (FFT). The top-K FFT coefficients were then stored. Through experimentation, we determined that only the top 10 coefficients are required to to regenerate the arrival process with a reasonable accuracy. The mean squared error (MSE) is shown in Table 3 which indicates that the error decreases as the number of FFT coefficients increases. Figures 10, 11, and 12 also show the regenerated arrival process for a varying number of FFT coefficients. Note that the regenerated process matches the diurnal pattern of the trace. In addition, we are also faithful in reproducing the spikes that occur every 30 minutes. These spikes probably occur as a result of users tuning into the VoD catalog after watching a broadcast program.

# of Coeff.	50	25	10
MSE	0.011065	0.011357	0.011549

Table 3: MSE for different FFT coefficients

4.2 Finite State Machine for Stream Control Events

For the purpose of modeling the stream control events, we initially investigated a discrete-time first-order finite state Markov process. A first-order Markov chain is entirely defined by the transition probabilities and an initial state distribution. Comparison with a real trace revealed a poor fit especially in terms of the session durations. To be specific, the session durations (sojourn times) for a Markov chain are geometrically distributed [16], whereas session durations in our traces are not. We first address the state transition probability estimation. The sojourn time estimation is then addressed in Section 4.3. The state transition probability matrix (P) is determined by counting the number of transition events from each state (e.g., PLAY to FASTFORWARD or PLAY to EXIT). We found that a single transition probability matrix does not adequately cover all scenarios. An examination of the state transition probabilities for weekdays and weekends found them to be sufficiently different to warrant separate transition probability matrices. We also examined transition probabilities for specific time-of-day effects (e.g., the busiest viewing period, typically called "prime time," is between 7 P.M. and 10 P.M.) but found that the variation was not large enough to warrant the additional complexity. We have thus chosen, in the interests of simplicity, to retain here only weekday and weekend models.

We observed that the primary determinant was the traffic intensity, rather than the transition probabilities for the stream control events over the busy hour. Although we could consider the effects of content genre and other environmental dependencies, we believe these have a second-order impact on the model.

An example of the estimated Markov chain is shown in Fig. 1. When users begin a VoD session by selecting a VoD video, the video starts to play automatically, initiating the START state. From this state, users may move to the other stream control states. For example, in Fig. 1, the user may go to FASTFORWARD with a probability of 0.23 or to PAUSE with a probability of 0.10. When they resume playing the video, they enter PLAY. While the state transition diagram has transitions from every one of the eight primary states to every other state, we only show the transitions with transition probability of 3% or more in Fig. 1. We refer the reader to the tables in [9] for the transition probability matrices for all the eight states obtained for both weekdays and weekends.

4.3 Modeling Sojourn Time Distributions

We consider two approaches for modeling sojourn time distributions. The first approach (SM), using a semi-Markov model, assumes that the event sojourn times are independent and focuses on fitting the sojourn time distributions for

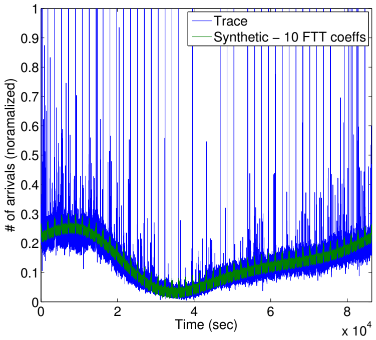

Figure 10: 10 coefficients

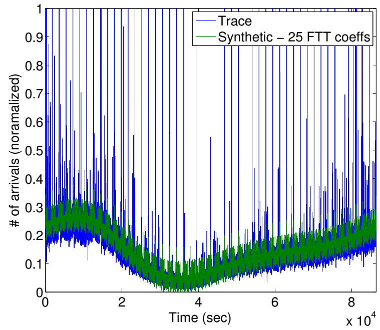

Figure 11: 25 coefficients

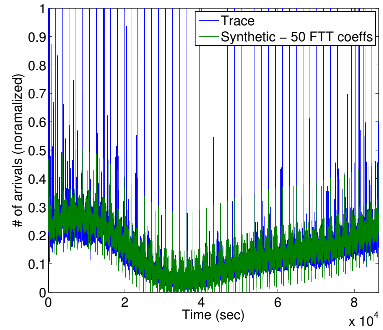

Figure 12: 50 coefficients

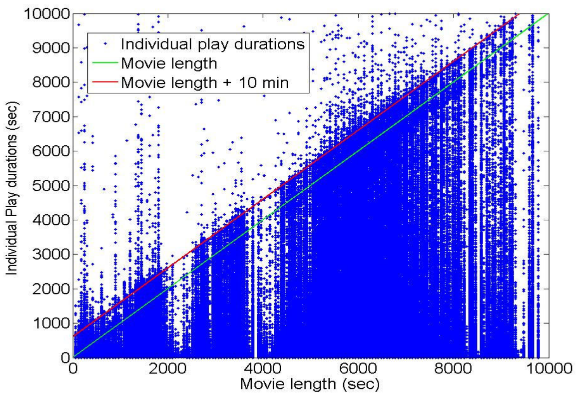

Figure 13: Play duration vs video length

each state of the FSM. In practice since the actual state sojourn times may not be independent, it may not match the distribution of the session durations. The second approach (CM) is a constrained model, the stick-breaking approach, and it imposes the session duration constraint.

4.3.1 Independent Sojourn Time Models (SM)

As motivated in the previous subsection we investigate the use of a semi-Markov model, where the sojourn time distributions can be freely chosen [6]. A semi-Markov model consists of an embedded discrete parameter Markov chain with a corresponding transition probability matrix and a set of conditional sojourn time distribution functions. In its most general form, the sojourn time distributions in state s are dependent on the next state to be visited. We begin with a restricted semi-Markov model, where the sojourn time distributions depend on the current state alone. After some experimentation (taking into account time of day and video length), we opted to normalize sojourn times with respect to video length and to create two separate models; one each for a typical weekend and weekday. Fig. 13 plots PLAY duration against video length. There are some PLAY events that persist longer than the video length (shown by the red line). This is an artifact of the IPTV system, which generates an automatic time-out after ten minutes when a viewer fails to exit after the video has ended, thus releasing this user from the VoD server and conserving resources. From Fig. 13 we observe that the PLAY duration (the dominant event) is seen to be proportional to the length of the video. This justifies normalization of the PLAY duration by the video length. For the sake of simplicity, we chose to normalize the sojourn times in the other states by the video length as well.We found that after normalization by the video length, the sojourn time in each state exhibited a weak day-of-week effect, adding further justification of our choice to normalize by the video length. (see Figure 14).

	Weekend		Weekday	
Operation	Mean	Variance	Mean	Variance
SKIP	53.56	1.48e5	39.29	9.53e4
FASTFORWARD	8.08	113.18	8.53	121.68
PLAY	305.22	8.37e5	195.32	4.92e5
REPLAY	66.61	1.53e5	54.80	1.147e5
REWIND	5.39	52.43	5.62	46.10
STOP	28.44	7.47e3	23.24	6.01e3
PAUSE	55.03	1.44e4	40.27	1.09e4
START	763.19	2.70e6	610.97	2.06e6

Table 4: Statistics of stream control durations (in seconds), for VoD.

We now examine the day-of-week effect on sojourn time. The mean values of the states PLAY and START are far greater than those of any other state; this is where viewers spend most of their time. Most of other states have much smaller means, indicating that users tend to switch out of them quickly. Table 4 shows the mean duration for each state. Note that the mean PLAY duration is smaller on weekdays than weekends (a 32% reduction). A smaller reduction of 19% is seen between weekend days and weekdays for the normalized PLAY durations. See Figure 14).

Thus, we have now reduced the problem to that of modeling the CDFs of the normalized sojourn times. Let $\{T_s(n),\ n = 0, 1, 2\ldots\}$ be the sequence of sojourn times in state s for a given user and let random variable T_s denote a generic sojourn time duration. Our objective is to model the CDF $F_s(\tau) := \Pr(T_s \le \tau)$.

Two commonly used models for characterizing heavy tailed distributions (HTDs) are the generalized Pareto and Weibull distributions. The maximum likelihood parameters for the Generalized Pareto function for shape and scale are $a = 2.174$ and $b = 0.0012$ respectively. Similarly, for the Weibull function they are $a = 0.0108$ and $b = 0.3391$. Fig. 14 and Fig. 15 shows that these standard HTDs are not a very close fit to the observed data.

To investigate further, we examine empirical CDFs (i.e., CDFs computed from data) for different states s in Fig. 16. In this figure, $\log(\tau)$ is plotted against $F(\tau)$ for each state F_s

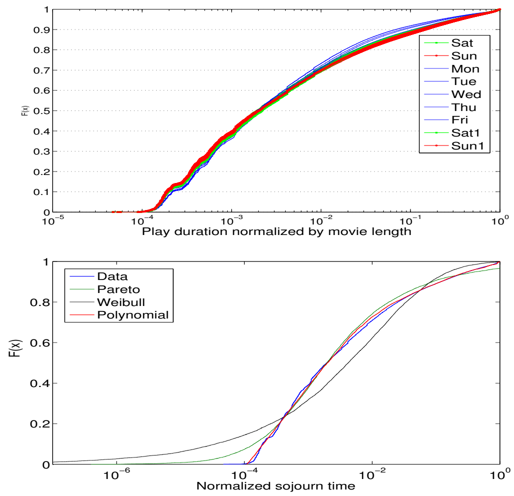

Figure 14: CDF of normalized play durations for VoD, different days and different distribution fits

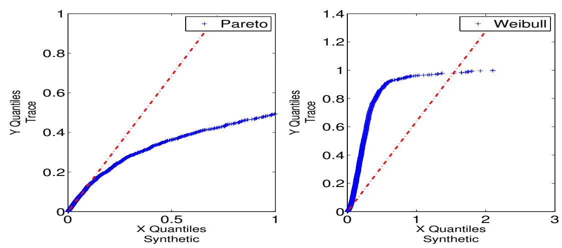

Figure 15: QQ plots for Pareto and Weibull generated synthetic traces

(in other words, this figure shows G_s, the inverse of the CDF on a logarithmic scale).[1] Fig. 16 suggests that we should try to represent $\log G_s(p)$ as a polynomial function of p.

We thus assume that the inverse CDF can be modeled by the $m+1$-term polynomial $\log \hat{G}(p) = \sum_{i=0}^{m} a_i p^{n_i}$ for given n_i, $i = 0, 1, \ldots, m$. For a given set of n_i's, we choose the parameter vector $\bar{a} = (a_0, a_1, \ldots, a_m)^t$ so as to minimize the $(N+1)$-point sum squared error (SSE)

$$SSE(N) = \sum_{p_i} (\log G(p_i) - \log \hat{G}(p_i))^2,$$

where $p_i = i/N$, $i = 0, 1, \ldots, N$ is a uniformly spaced set of points in the interval $[0, 1]$ at which the empirical inverse CDF is sampled. From linear estimation theory, the vector parameter $\bar{a}$ that minimizes the SSE is given by

$$\bar{a} = (U^t U)^{-1} U^t g, \qquad (1)$$

where U is a $(N+1) \times (m+1)$ matrix, whose ith column is $(p_0^{n_i}, p_1^{n_i}, p_2^{n_i}, \ldots, p_N^{n_i})^t$, $i = 0, 1, 2, \ldots, m$, and g is the column vector $(\log G(p_0), \log G(p_1), \ldots, \log G(p_N))^t$, which is obtained from the data.

We experimented with several choices for the basis functions and used the quantile-quantile (QQ) plot as a guide to the quality of the fit. Our search over $n_i = i^k$, $k = 1, 2, ..., 6$ revealed that $k = 3$ is optimal, and we provide some evidence to support this in the QQ plots shown in Fig. 18 and Fig. 19.

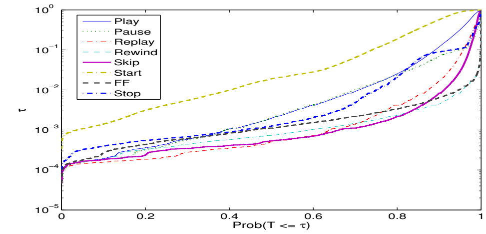

Figure 16: Sojourn time inverse CDFs

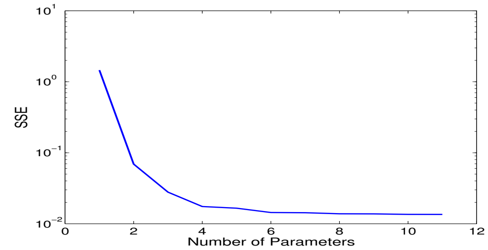

Figure 17: SSE as a function of number of parameters, m, for the Fast-Forward duration CDF.

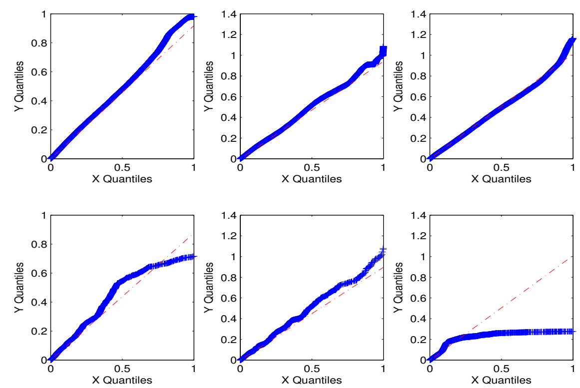

Figure 18: Various QQ plots. Horiz. axis: actual data. Vert. axis: synthetic data. Top row: play; bottom row: Pause. Left to right $(m, k) = (6, 3), (11, 3), (6, 1)$, respectively.

For $n_i = i^3$, the SSE is plotted as a function of m in Fig. 17. Even though this plot suggests that $m = 6$ parameters are sufficient, the QQ plot suggests that the fit is not sufficiently linear until $m = 11$.

We also examined the correlation structure of the collected data. The first-order correlation coefficient, i.e., the correlation coefficient of successive sojourn times, is shown in Table 5. The correlation coefficients are seen to be small, though not insignificant, especially in the PLAY state. While we could choose to model the sequence of sojourn times as a correlated autoregressive process, we would have to give up control over the marginal CDF of the generated process. Given the small correlation coefficients, we opted for simplicity and chose an independent identically distributed process.

Finally, we remark that constructing a fit for the inverse CDF, rather than the CDF, has an added advantage that a random variable T with the desired distribution can be

[1] To work around the problem that F_s will not have an inverse when the underlying distribution has nonzero mass at a point, we define $G_s(p) = \arg\min_x \{F_s(x) \geq p\}$.

Play	Pause	Replay	Rew	Skip	Start	FF
0.21	0.21	0.05	0.10	0.16	0.14	0.09

Table 5: First-order corr. coeff. for the sojourn time of each state.

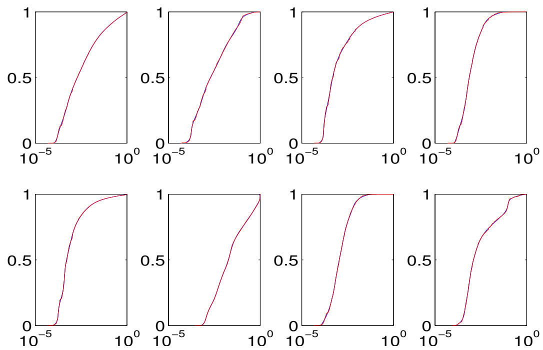

Figure 19: Fitted Distributions, $m = 11$ parameters. Top, left to right: play, Pause, Replay, Rewind. Bottom, left to right: Skip, start, FastForward, stop.

obtained by warping U a random variable uniformly distributed on $[0, 1]$ using the formula $T = \exp(\sum_{i=0}^{m} a_i U^{n_i})$.

We computed the mean, variance, third moment, and median of all state distributions to compare the accuracy of the various fits. As seen in Table 6, the family of polynomial exponential fitting functions performs better than the standard HTDs considered here with respect to these metrics. Table 6 shows that the polynomial fitted distributions fit the third moment and the median of the data set better, unlike the generalized Pareto and Weibull functions. The polynomial distribution has the closest fit and has an accuracy for all metrics to within 1% of the data. Additionally, Fig. 14 shows the closeness of the polynomial fit when compared to the real trace's normalized CDF.

	Trace	Pareto	Weibull	Polynomial
E[X]	0.0499	0.0269	0.0324	0.0498
Var[X]	0.0220	0.0090	0.0068	0.0224
E[X^3]	0.0137	0.0051	0.0030	0.0143
Median	0.0019	0.0018	0.0043	0.0019

Table 6: Comparison of statistics of trace data and fitted distributions for play state.

The state transition matrices and polynomial coefficients are tabulated in [9]. Fig. 20 shows the Q-Q plots for the synthetically generated data and the normalized trace data for the sojourn times for each of the VoD stream control states. All the state durations are close to the reference line, indicating a good match between the observed and the modeled data.

4.3.2 *Stick-Breaking Sojourn-Time Models (CM)*

The modeling approach in Sec. 4.3.1 assumes independent sojourn times for each stream control event. In reality, event sojourn times are not independent—e.g., a REWIND event cannot have a duration that exceeds the total sum of the preceding PLAY events. A problem with the independence assumption is that the session duration distribution for the synthetic trace would be a sum of independent random variables and this is in general different from the actual session duration distribution. We show in Fig. 21 the cumulative distribution of session durations observed on a weekday and a weekend in real traces and through the approaches in Sec. 4.3.1. It is clear from the figure that the FSM approach generates a considerable number of short sessions and compensates by generating a few very long sessions.

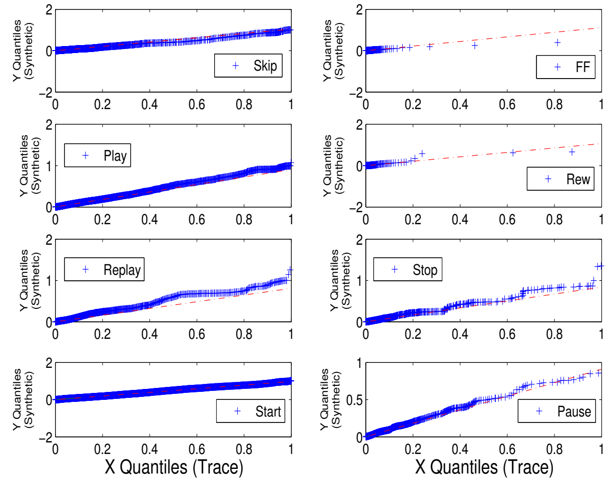

Figure 20: Q-Q plots, weekend (normalized)

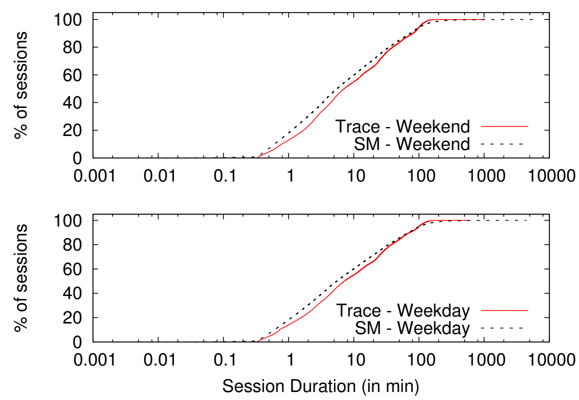

Figure 21: CDF of session duration for real- and SM-based trace on a weekday and weekend.

To rectify this situation, we present an approach that preserves the overall session duration distribution. The main idea is to model the fraction of the session duration taken up by the sojourn time in each state, as well as the overall session duration. Since the state sojourn times must add up to the overall session duration they cannot be independent, so this model does capture some of the subtle dependencies between the state sojourn times. Later in this section, we address the issue of joining together the FSM and state sojourn time models.

Models for non-negative random variables with a specified sum have been studied in the probability theory literature [14] and more recently have found application in machine learning and pattern recognition. If the sum of the random variables is unity, then a probabilistic model for the random variables is equivalent to constructing a measure on the space of probability distributions. In the machine learning literature, the reason for the interest in such random measures comes from a desire to determine a maximum likelihood estimate of the prior distribution given a set of observations, see e.g., [2] for an estimation problem related to Gaussian mixture models.

The approach that we take here is to model the fraction of the residual session duration that is occupied by each state. This approach, referred to as the stick-breaking approach [14], goes back at least as far back as [12]. Let T_s denote the random session duration, and let T_i denote the random duration of the ith event, $i = 1, 2, \ldots, K$ ($K = 7$ in this case). Thus $T_s = T_1 + T_2 + \ldots + T_K$, where K is the number of distinct types of stream control events. We model the distributions of the random variables $X_1 = T_1/T_s$, $X_2 = T_2/(T_s - T_1)$, and in general $X_i = T_i/(T_s - \sum_{j=1}^{i-1} T_i)$, $i = 1, 2, \ldots, K$. Thus X_i is the fraction of the 'remaining' session duration occupied by the ith event, and $X_K = 1$. We also model the distribution of T_s. Let p_i be the modeled pdf of X_i and let p_s be the modeled pdf for T_s. An individual session is then generated by picking the fraction X_i independently and according to p_i, $i = 1, 2, \ldots, K - 1$, $X_K = 1 - \sum_{i=1}^{K-1} X_i$ and T_s according to p_s. The individual session durations are then given by $T_1 = X_1 T_s$, $T_2 = X_2(T_s - T_1)$, and in general $T_i = X_1(T_s - \sum_{j=1}^{i-1} T_j)$. This approach preserves both the total session duration as well as the fraction of a residual session occupied by each event.

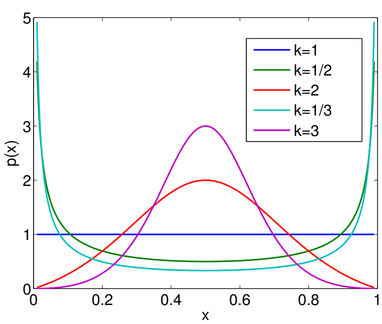

Figure 22: The pdf of X_i for different values of the shape parameter for the underlying Weibull distribution. Heavy tails lead to cup shaped distributions.

We have thus reduced the problem of modeling the state sojourn durations to that of modeling the fractions X_i, $i = 1, 2, \ldots, K$. In most machine learning applications, the observed data comes from a Bernoulli process or Bernoulli scheme, and the estimation problem is to determine the most likely prior probability distribution, given the observed data. The beta distribution is a natural class of prior distributions in this setting [14]. The beta distribution is a parametric family of distributions on the open unit interval given by

$$p(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}. \tag{2}$$

which are well suited to modeling the X_i's. The mean and variance of the beta distribution is given by $a/(a+b)$ and $ab/(a+b)^2(a+b+1)$, respectively, and the distributions are cup-shaped for a, b in the interval $(0, 1)$.

Compared to the standard machine learning setting, our fractions X_i arise from competing sojourn times and it is not obvious that beta random variables will be useful. Some insight into the modeling of the random variables X_i can be obtained from the following toy example. Consider a pair (T_1, T_2), of independent identically distributed sojourn times with the Weibull distribution,

$$p(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} U(x) \tag{3}$$

where $U(x)$ is the unit step function. Let $X_1 = T_1/(T_1+T_2)$. The pdf of X_1 is plotted in Fig. 22 for different values of the shape parameter k. It is surprising that the observed pdf's are close in distribution to beta distributions. More interesting is the fact that heavy tails for the Weibull distribution, $k < 1$, lead to cup shaped distributions for the X_i's. In other words, *heavy tails lead to extreme ratios*.

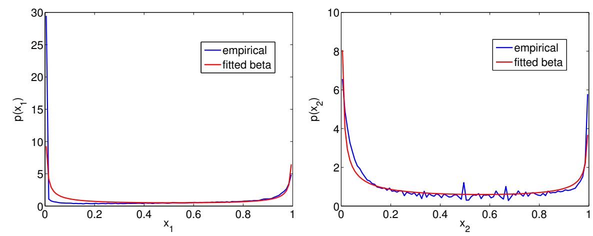

Figure 23: Empirical and fitted beta distributions for X_1 (left) and X_2 (right).

While the above toy example was based on the Weibull distribution for state sojourn durations, it is seen in Fig. 23 that the X_i's obtained from the trace data are also well modeled by the beta distribution, and in fact have the distinctive cup shape that we observed when the Weibull distribution has heavy tails. We also show in Fig. 23 the fitted beta distributions in which the parameters a and b are estimated so as to match the empirical mean and variance of X_1, $i = 1, 2, \ldots, K-1$. Thus, estimated parameters $\hat{a}$ and $\hat{b}$ are given by $\hat{a} = \mu\gamma$, $\hat{b} = (1-\mu)\gamma$, with $\gamma = \frac{\mu(1-\mu)}{\sigma^2} - 1$. The matching between the empirical and fitted distributions is seen to be quite good.

We now address the problem of joining the stick-breaking sojourn times with the FSM model. There are two issues to be addressed: (i) the FSM might visit a state several times, while the stick-model produces a single number for the aggregate time spent in that state, and (ii) a sample path through the FSM might exclude some states for which the stick-breaking model generates a positive fraction X_i. Setting this X_i to zero will perturb the resulting session duration distribution.

To address (i) the fraction X_i is divided equally by the number times the FSM visits state i. To address (ii) we generate a number of FSM sample paths, as well as a number of sample fractions using the stick breaking model. A matching algorithm is then used to match FSM sample paths and the stick-breaking outcomes, to yield the CM model.

4.4 DVR Model: Highlights

Much of the DVR modeling issues are similar to those encountered while modeling VoD. For example, the DVR arrival process is quite similar to the VoD arrival process – a superposition of a bursty and a smooth component with a well-defined diurnal variation. Hence, we only emphasize the key differences in this section. Figs. 24 and 25 show the normalized sojourn time for VoD and DVR respectively for a weekend day. The estimated state transition matrix for the DVR state machine revealed a few important differences compared to the VoD state machine. The transition probabilities from the START to EXIT states are 46% and

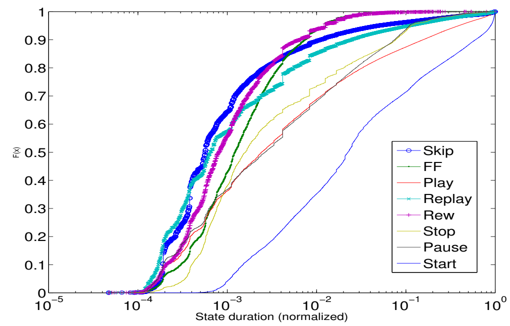

Figure 24: All stream control CDFs, weekend (normalized) for VoD

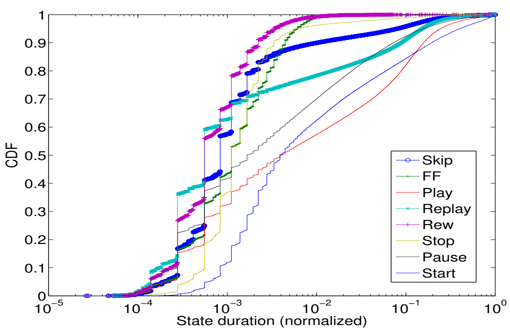

Figure 25: All stream control CDFs, weekend (normalized) for DVR

17% for VoD and DVR respectively, indicating less use of stream control during VoD viewing. Second, the transition probabilities from PLAY to FASTFORWARD are 41% and 55% for VoD and DVR respectively. The higher probablility of FASTFORWARD in the DVR model is probably explained by the efforts of viewers to avoid the advertising that forms such a large part of broadcast television programming. Similarly, SKIP has a higher probability in the DVR model, presumably also due to users' efforts to avoid ads.

We have also developed DVR stream control models for both the weekend and weekday. We refer the reader to our technical report for the detailed model parameters [9].

5. EXPERIMENTAL RESULTS

One of the goals of our work is provide the ability to generate synthetic traces that effectively capture users' behavior in a large-scale VoD system. In this section we validate the accuracy of our models by generating completely synthetic traces and comparing it to real traces using a custom event-driven simulator. Stream control events affect how much of a video is watched. This directly translates into how much load each session places on the server. As a result, we use server load as our primary metric of comparison. We use the simulator to accurately capture the effects of stream control events on server load. However, we also examine the generated traces for other attributes and show that our models are able to successfully capture the important aspects of users' VoD viewing sessions.

5.1 Synthetic Trace Generation

We used our models to generate completely synthetic traces for a given subscriber population. The process starts with the generation of a stream of session start requests. There is variability in the number of arrivals, both on week days and weekends, even for the same number of subscribers over the 2 year period. This may be due to a variety of societal factors (seasonal viewing patterns, TV programming and new content generation cycles etc.) As a result, we compute the mean of all traces (normalized by their respective number of subscribers for those days). We use the process described in Section 4.1 to generate a synthetic trace, and scale our modeled arrival process for a given subscriber population.

Once we have the arrivals, we then model the session behavior using either the SM or the CM model. In the case of SM, for each session, we used the stream control model to generate a sequence of stream control events and their sojourn times. Since each FSM produces events whose lengths are normalized by movie length, the individual state sojourn time durations generated by each FSM are then scaled by the movie length. The movie length is drawn from a distribution fitted (using the Kaplan-Meier technique) to the empirical movie length distribution shown in Figure 5.

In the case of CM, given a realization of X_i, the fraction of the remaining session duration occupied by the ith event, we generate independent sequences of stream control events by using the FSM. The output of the FSM gives us a sequence of events in a particular order from the START state to the EXIT state. We divide the X_i's equally by the number of events for state i to allocate the individual time spent per state. Finally, for a randomly chosen session duration (obtained from an appropriate session duration distribution), the X_i's are rescaled by the session duration. This allows us to retrofit the session durations from the stick-breaking model to the sojourn times with the FSM.

5.2 Experiment Setup

We validated the accuracy of our models, by running the synthetic and actual traces through the simulator. Our simulator faithfully models the message exchanges between a client (i.e., set top box) and a video server with an abstract model of the network between the client and the server.

Modeling Videos. We model videos as consisting of a sequence of 2-second chunks. In the absence of stream control events, these chunks are requested and played out in sequence. This is similar to the techniques used in many P2P systems [8, 15] and in some centralized approaches (e.g., HTTP Live Streaming [19]). With stream control, clients jump to different parts of the video depending on the event, and then request the chunks sequentially from that point on. We assume that FASTFORWARD and REWIND are implemented using "trick streams". Trick streams essentially are alternate video files that encode only certain key frames of the video and provide the impression that the video is being played faster.

Client modeling. Clients initiate requests for videos based on the input file. On receiving a PLAY request, the client sequentially requests one chunk at a time until it executes a stream control event. We assume that a SKIP takes the viewer 30 seconds into the video, while a REPLAY results in a 7-second jump backwards. When clients execute one of these operations, the existing transfer is aborted and a request is sent out to the server for the new chunk or a trick stream.

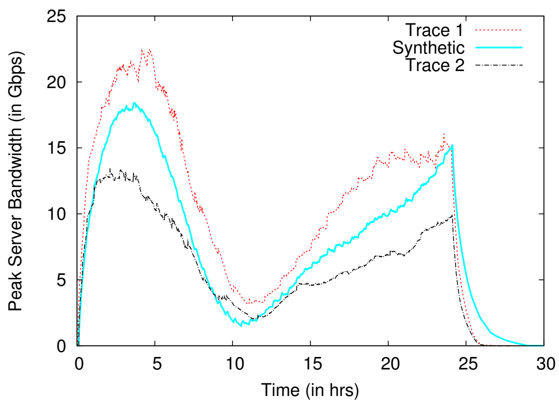

Figure 26: Peak server load generated by two different real traces and the pure synthetic trace, for a commensurate number of subscribers.

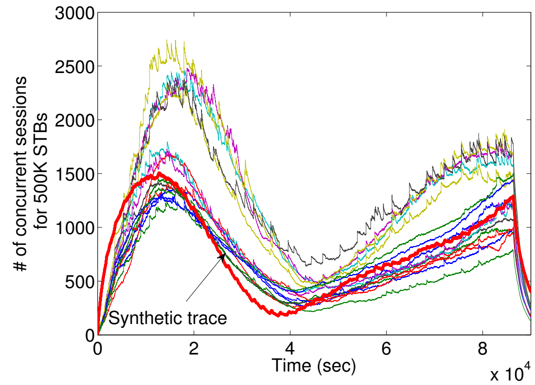

Figure 27: # of concurrent sessions (all Tuesday traces and synthetic trace) scaled for a population of 500K

Each client is assumed to have a local disk which it uses to store the data being transferred by the server for the duration of the video (after which it is deleted). The client waits for four seconds of data in its playout buffer before starting to play the video. Thus, when a client performs a SKIP or a REPLAY, the client waits until sufficient data is buffered before continuing to play. Finally, the client tries to follow the state transitions faithfully and ignores events that it encounters in the trace that cannot be executed in its current state.

Server Modeling. The server in our simulator responds to requests for chunks and trick streams by delivering them as a unicast stream. The server keeps track of the existing transfer for each client. Note that the server only gets chunk requests and is oblivious to the specific stream control operation performed. This behavior is not very different from what many commercial VoD systems implement [7]. If it receives a request for a new chunk or trick stream before the end of an existing transfer, the server aborts the existing transfer, and starts serving the new request. We assume that all video streams and trick streams are encoded at 2 Mbps and that the server transfers the video 10% faster (2.1 Mbps) than the playout rate to accommodate for transient network conditions.

5.3 Server Load: Synthetic vs. Real Traces

For our main result, we compare the load at the server due to real and synthetic traces. Nominally, we generated a synthetic trace for a weekday using a subscriber base of ~3 million. We compare this to the server load due to real traces with a commensurate number of subscribers.

We present the result in Figure 26. The plot shows the peak server bandwidth (measured every minute) over time for the two real traces and the purely synthetic trace. There are a few observations we make. First, the synthetic trace nicely captures the diurnal changes in load and resembles the pattern observed in the real traces. Next, for the same number of subscribers and on the same day of the week, the real traces can actually generate significantly different load. This is because the number of *concurrently active* users is different at these different times. Finally, we see that the load due to the synthetic trace falls between the two real traces. This is not unexpected because we model the session arrivals to be an average of the arrivals across the different days (refer Section 5.1). We validate this next.

5.4 Accuracy of Session Arrival Model

In the first result, we attributed the difference in server load between the synthetic trace and real traces to the variability in load even in the real traces. To validate this, we plot the number of concurrent sessions for all the 16 Tuesdays (spanning 2 years, with varying number of subscribers) in our DS1 data set. To remove the inherent scaling issues with a growing subscriber base, we rescaled each trace individually to a population of 500K subscribers. We show the number of concurrent sessions for the 16 Tuesdays in Figure 27.

The peak number of concurrent sessions varies widely from 1200 to 2700 during the busy hour. We did not observe any distinctive correlation (e.g., based on seasons) between the different Tuesdays. We also plot the concurrent sessions for the purely synthetic trace in the plot using the thick red line. The synthetic trace matches the diurnal variation of the empirical traces. We also observe that the synthetic trace falls within the upper and lower bounds of the number of concurrent sessions. This is important: given the variability, we cannot exactly match the session arrivals, but this result shows that model is able to generate a representative set of session arrivals. In addition to the number of provisioned subscribers, we also examined the variability if we normalize by just the number of *active* subscribers (i.e., makes at least one request on a given day). Our results (not shown) indicate very similar variability across the 16 days.

5.5 Comparison of Stream Control Models

In this experiment we study the accuracy of the two stream control models we have proposed. We generate a semi-synthetic traces using each of the two stream control model approaches (SM, CM), for both a weekday and a weekend and compare them against the representative real traces. In order to eliminate any effects of the session arrival generation process, we use the arrivals from the real traces, but only generate the stream control events and their durations using the stream control models (this is the reason we say these are a semi-synthetic traces). We present the results both for a weekday and a weekend day in Figures 28 and 29 respectively.

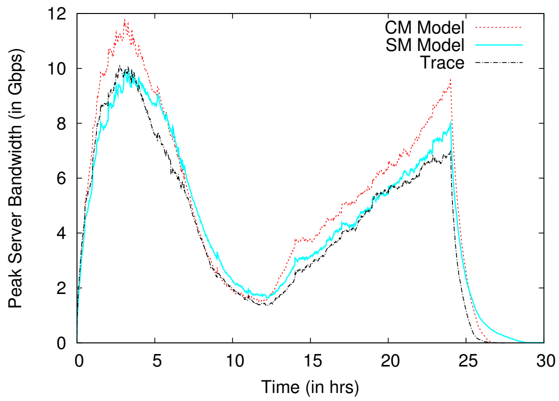

Figure 28: Peak server load on Friday generated by real and synthetic (SM and CM) traces.

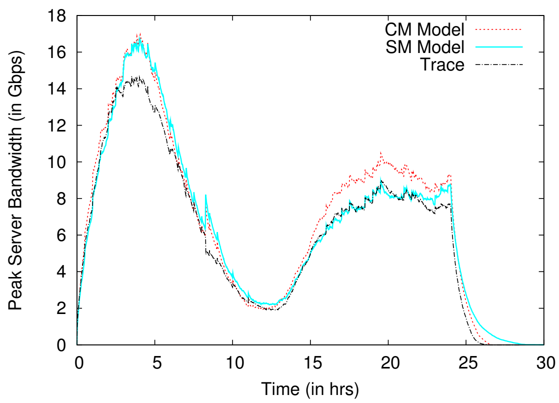

Figure 29: Peak server load on Saturday generated by real and synthetic traces (SM and CM).

The plots shows that our modeling of stream control functions accurately captures the load at the server. In particular the plots show that the SM-based model almost matches the load at the server at all times indicating that it is able to accurately capture the effects of stream control. The stick breaking (CM-model) approach, while performing well, overestimates the load at the server. This result goes to show that it is not only important to generate accurate session durations; it is also important to get the number and sequence of stream control events correct.

5.6 Stream Control Events in a Session

We then compare the number of stream control events observed in the real trace to that generated in the synthetic traces. Specifically we compare the average number and average duration of stream control events per session for both a weekend day and weekday. We report the results in Table 7.

The SM-based approach, on average, generates a similar number of stream control events and durations compared to the CM approach. In particular, we see that the CM approach overestimates the REWIND and FASTFORWARD durations, which adds load to the server. On the other hand, it underestimates the PAUSE duration which reduces load on the server. This explains the difference in the peak server bandwidth observed in Figures 28 and 29 between the SM and CM approaches. The reason for the inaccuracies in event durations generated by the CM model is because of error accumulation in the regenaration process, where beta random variables are successively subtracted off to generate the time intervals associated with each event. For this reason, we believe it is better to estimate the beta random variables in an increasing order of the mean value of the event durations and then reconstruct the sojourn times for each of the states from these generated beta random variables.

It is observed that the SM approach results in very high variability in the session durations, while the trace using CM does not. For example, consider the total time spent by the user in the play state. The real trace had a standard deviation of 1856.46 for the weekday and 1940.22 on the weekend day. The SM model had 2928.95 and 2892.78 for the weekday and weekend respectively. But, the CM model had 1744.70 and 1748.08 for the weekday and weekend, matching the real trace more closely. Thus, while the SM model captures the average durations nicely, CM captures the variability better.

5.7 Client Session Interruptions

As clients perform stream control operations, they may experience interruptions in their viewing of the video until enough data is buffered in the play-out buffer. In this experiment we characterize the interruptions experienced by clients in the real trace in comparison to the synthetic trace. Note that the exact number of interruptions and their duration depends on the specific client implementation, the chunk size used, etc. Our goal here is to compare the viewer experience with the real trace and the synthetic traces with one example implementation.

For each stream control operation, we identify if that portion of the video needed is locally available in the client buffer. If not, we request for that portion from the server. We assume that a minimum of 4 seconds of video has to be buffered before the video can be displayed. If the desired portion is not available, we count it as an interruption and measure the time that the viewer has to wait before the video starts playing again.

Data	% of interr. sessions	Avg. interr. per session	Avg. Duration of interr.
Syn. CM	41.31	3.17	4.98
Syn. SM	38.11	2.51	4.76
Real Trace	37.45	3.47	5.17

Table 8: Session interruption statistics

Table 8 presents the statistics related to interrupted sessions for the real trace as well as the synthetic traces generated using the two alternate models (the CM model and the SM model). The results show that the fraction of sessions experiencing interruptions with the synthetic traces is comparable to that with the real trace, with the SM model doing slightly better than the CM model. However, among the interrupted sessions, the CM model more accurately captures the average number of interruptions per session (3.17 for CM model vs. 3.47 for the real trace) than the SM model (2.51 interruptions per session). The CM model also captures the average duration of interruptions better than the SM model. However, the differences between the two models, for this metric, are not statistically significant enough to advocate one model over the other.

	Weekend						Weekday					
	Avg. Count			Avg.Duration			Avg.Count			Avg.Duration		
State	Real	SM	CM	Real	SM	CM	Real	SM	CM	Real	SM	CM
Play	3.13	3.62	4.73	1163.24	1108.06	1120.03	2.89	3.31	4.61	1232.31	1178.51	1120.75
FastForward	1.93	2.31	3.02	15.78	38.98	90.48	1.55	1.80	2.60	12.38	34.78	89.16
Rewind	0.49	0.63	1.17	2.56	9.89	68.42	0.50	0.64	1.20	2.56	9.89	68.42
Pause	0.71	0.89	1.27	63.44	84.65	37.92	0.70	0.90	1.38	61.16	94.21	37.42
Skip	1.61	1.90	2.82	76.21	138.90	37.38	1.32	1.49	2.50	69.60	129.28	38.47
Replay	1.06	1.39	2.28	62.63	128.92	17.31	0.95	1.19	2.20	62.07	126.68	17.69

Table 7: Breakup of stream control events on weekdays and weekends.

6. CONCLUSIONS

We set out to understand and model interactive user behavior in an IPTV environment, particularly its impact on system resources. We modeled user interactivity based on traces of user actions captured from a nationally deployed, operational system. We developed a video request arrival process model and two comprehensive stream control models. The frequency domain (FFT) based arrival process model faithfully captures the diurnal pattern while also preserving the periodic bursty nature of the traffic with only a few parameters. The stream control model consists of a finite state machine and two alternative models for determining the state sojourn times. The first alternative (SM model) seeks to preserve the sojourn time distribution for each state. The second alternative (CM model) puts a greater emphasis on preserving the overall session duration distribution.

To establish the validity of our overall models for user interaction, we generated completely synthetic traces of user interactions and fed both the synthetic trace and the real trace to a simulator of the IPTV VoD system. We compared the resulting bandwidth requirement on the VoD server and showed that the synthetic load from our model achieves a very good match to the load imposed by the real traces.

When a comparison is made based on the peak server bandwidth, the SM model provides a closer match to the empirically observed peak server bandwidth as compared to the CM model. However, the CM model captures the standard deviation of the session durations more accurately. We feel that while both alternatives are valid the SM model gives us a better estimate for provisioning capacity, a task of interest to providers.

7. REFERENCES

[1] A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *Web Mining Workshop*, pages 33–40, April 2001.

[2] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. *Hierarchical Topic Models and the Nested Chinese Restaurant Process*, volume 16 of *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[3] P. Branch, G. Egan, and B. Tonkin. Modeling interactive behavior of a video based multimedia system. In *Proceedings of IEEE ICC*, pages 978–982, 1999.

[4] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the Worlds Largest User Generated Content Video System. In *Proceedings of ACM IMC*, 2007.

[5] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatrianin. Watching Television Over an IP Network. In *Proceedings of ACM IMC*, 2008.

[6] A. Csenki. Dependability for systems with a partitioned state space. *Springer-Verlag*, 1994.

[7] P. Diminico, V. Gopalakrishnan, R. Jana, K. Ramakrishnan, D. Swayne, and V. Vaishampayan. Capacity Requirements for On-Demand IPTV Services. In *Proceedings of COMSNETS*, January 2011.

[8] V. Gopalakrishnan, B. Bhattacharjee, K. K. Ramakrishnan, R. Jana, and D. Srivastava. CPM: Adaptive Video-on-Demand with Cooperative Peer Assists and Multicast. In *Proceedings of IEEE Infocom*, April 2009.

[9] V. Gopalakrishnan, R. Jana, K. Ramakrishnan, D. Swayne, and V. Vaishampayan. Understanding Couch Potatoes: Modeling Interactive Usage of IPTV at Large-Scale. Technical report, AT&T Labs Research, May 2011. www.research.att.com/∼kkrama/papers/stream-control.pdf.

[10] V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, D. F. Swayne, and V. A. Vaishampayan. Characterizing Interactive Behavior in a Large-Scale Operational IPTV Environment. In *IEEE Infocom (mini-conference)*, March 2010.

[11] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang. The stretched exponential distribution of internet media access patterns. In *Proceedings of PODC*, 2008.

[12] P. Halmos. Random Alms. *The Annals of Mathematical Statistics*, 15:182–189, 1944.

[13] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A measurement study of a large-scale P2P IPTV system. *IEEE Transactions on Multimedia*, 9(8):1672–1687, Dec 2007.

[14] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the Americal Statistical Association*, 96:161–173, March 2001.

[15] V. Janardhan and H. Schulzrinne. Peer assisted VoD for set-top box based IP network. In *SIGCOMM workshop on P2P-TV*, Aug 2007.

[16] L. Kleinrock. *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, 1975.

[17] E. Manavoglu, D. Pavlov, and C. L. Giles. Probabilistic user behavior models. In *Proceedings of IEEE ICDM*, Nov 2003.

[18] S. Mongy, C. Djeraba, and D. Simovici. On clustering users' behaviors in video sessions. In *Proceedings of ICDM*, 2007.

[19] R. Pantos and W. May. HTTP Live Streaming. http://tools.ietf.org/html/draft-pantos-http-live-streaming-06.

[20] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q.Zhao. Modeling user activities in a large IPTV system. In *Proceedings of ACM IMC*, 2009.

[21] N. Vratonjic, P. Gupta, N. Knezevic, D.Kostic, and A. Rowstron. Enabling DVD-like Features in P2P Video-on-demand Systems. In *Sigcomm workshop P2P-TV*, Aug 2007.

[22] D. Wu, Y. Liu, and K. Ross. Queuing Network Models for Multi-Channel P2P Live Streaming Systems. In *Proceedings of IEEE Infocom*, 2009.

[23] H. Yin, X. Liu, F. Qiu, N. Xia, C. Lin, H. Zhang, V. Sekar, and G. Min. Inside the Bird's Nest: Measurements of Large-Scale Live VoD from the 2008 Olympics. In *Proceedings of ACM IMC*, 2009.

Summary Review Documentation for

“Understanding Couch Potatoes: Modeling Interactive Usage of IPTV at large scale”

Authors: V. Gopalakrishnan, R. Jana, K. Ramakrishnan, D. Swayne, V. Vaishampayan

Reviewer #1

Strengths: The paper is based on a significant data set. The authors have developed one model for the arrival process and two models for the user interaction. They discuss the pros and cons of each of the user interaction models in re-generating interesting properties of the original data set. The results show that the derived models are able to approximate the workload imposed on the server in terms of bandwidth.

Weaknesses: I really appreciate the effort that the authors have put in clearly describing the objectives and analyzing the collected data sets. However, I was left wanting more in terms of validation of the synthetic trace. I may be wrong but I would expect that the interesting aspects of interactivity on the service itself are not restricted to bandwidth alone. I would love to have seen some more complex metrics, such as computational overhead on the server, the number of sessions that are aborted, the number of times a client would need to rebuffer or the delays that the user would experience based on the trace and the model - these metrics are more meaningful when it comes to designing a well performing service - not bandwidth.

Comments to Authors: This is a very complete paper. Based on an authoritative dataset the authors are able to study the interactions of users with an IPTV service. They can derive the different states that a user consuming IPTV may be found at, and derive the associated state and transition probabilities. They can further model the request arrival process. The intellectual exercise then is focused on identifying how to model the sojourn and transition probabilities of the Markov model. The authors make the case that we would need to derive two sets, one for the weekdays and one for the weekend. They then come up with two proposals, one that is able to faithfully replay the transitions from one state to the other, and the second which aims to result in an accurate distribution of the overall session duration.

My main concern with this work is that the final validation of the model leaves me wanting more. When one targets the efficient design of an IPTV system the metric of interest is not really server bandwidth. And the authors test the accuracy of their modeling contrasting the server bandwidth from the trace to that of the model. They further provide an evaluation of the stream control events in Table 6, but frankly the results there are far from conclusive. The errors made by both SM and CM are actually non-negligible.

More importantly, I would claim that there are other evaluation metrics that would be more meaningful. What is the computational overhead on the server, what are the delays faced by the user due to seek times on the server seeking for the appropriate content to stream, how many times did clients have to rebuffer? These would be more meaningful in the study of a system and should be able to be captured by a synthetic model. I do not see that in this paper. Therefore, its potential uses are restricted in my mind.

More detailed comments:

- It is so hard to understand what data the authors actually use. In Section 1 it appears we are talking about 16 weeks but those weeks are never mentioned. Later there is a mention to 150 days (Section 4.1). In Section 3.2 the authors talk about processing the next 60 days!!! How can this be the same dataset? Please be precise.
- There is an assumption that control events create server overhead but this is never proven.
- Table 5 is nice
- In Section 5.2 you talk about clients not following the state transitions faithfully. Why is this the case? What causes those?
- It is not clear which model is used for Figures 23 and 24. Is it the combined model?

Reviewer #2

Strengths: This is a thorough modeling paper, it takes into account various parameters concerning the sojourn time, transition probabilities, request arrival process, etc. It also uses an impressive data set to validate the model and the results show that the model can faithfully regenerate realistic loads.

Weaknesses: The main problem I have with this paper is the very fact that it is a modeling paper since I am not sure what will be its real applicability to inferring the real end2end user experience. The paper models a small part of the whole IPTV ecosystem, e.g. the arrival rate of users and its bandwidth. While this can be useful to model single components of the system, e.g. the server capacity, it still misses other important factors such as STB buffer sizes, routing policies, or other intermediate nodes in the network such as CDN nodes, transcoders in the injection points, etc.

Assuming that one would ignore all those other factors and just focus on those modeled in the paper, the authors do not show either why such a model is better suited than a much simpler one to dimension the system, e.g. a model that uses the mean and the peak capacity. Or for that matter what factors benefit the most by the complexity of this model.

All in all, it is a nice modeling paper but lacks enough results to show that it will have a significant impact in capturing the fundamentals of a real IPTV VoD system.

Comments to Authors: None.

Reviewer #3

Strengths: The problem addressed is an important one. The authors used (large amount) real data to develop and validate their results.

Weaknesses: I would like to better understand the independence, or lack of it, between the data used in model derivation and the trace data used for validation.

After deriving the models, the paper stated that "Nominally, we generated a synthetic trace for a weekday using a subscriber base of ~3 million.", but has this same data been used as real traces in the experiment evaluation? A clear explanation would be helpful.

Comments to Authors: none.

Reviewer #4

Strengths: The model developed in this paper can be useful for other research related to IPTV system. The derivation of the model is done carefully and systematically. The results from the model validation seem promising.

Weaknesses: The presentation of the paper needs more clarification.

Comments to Authors: The paper is well written and interesting to read. The derivation of the model is well supported by the data and analysis. Although there is room for improvement in terms of the accuracy, the results from the validation seem to be promising. Some specific comments for improvements are provided below.

Section 3.1: Is there any explanation/hypothesis why the number of events in the longer videos is smaller than in the 30-60 minute ones?

Section 3.2: What are the 5 clusters? Are those the same as in Table 1?

Section 3.3: How do you do the simulation? Is it the same simulation as in Section 5.2?

Section 4.3.1:
- Logically some events, e.g., skip or stop, might not need to be normalized. This could be determined by looking at the correlation similar to what has been done with the PLAY duration. Have those correlations been considered?
- Paragraph 2, it was mentioned that PLAY, START, and PAUSE are longer than other events, but from Table 3, REPLAY is longer than PAUSE.
- Could you explain why the time for SKIP is quite long? It is quite surprising because I think SKIP is simply a short jump to some specific point of the content and then immediately go back to PLAY state. It might also be a good idea to give a little more detail about each operation in Section 3.

Section 5.2: In the client modeling part, what are the differences between the "SKIP" and "REPLAY" operations you assume and the real ones?

Section 5.6:
- For comparison, it would be interesting to see the standard deviations as well.
- The results show that the average values for some states are still quite different from the real traces. This might be further improved in future work.

Reviewer #5

Strengths: - Understanding and modeling content requests and user interaction in IPTV is a timely issue. This paper can be a good reference. A large dataset in an operational network was used to derive the model.
- Solid analysis and modeling for both the arrival process and the stream control model. They also take into account the user behavior during weekdays and weekends.
- The authors also provide an appendix of results that researchers can use to validate their protocols.
- The paper is well written.

Weaknesses: I am a little bit puzzled with the second to last paragraph in page 2 where the authors describe what is the input data to derive and validate the model.

Comments to Authors: - The authors should elaborate more on the input data they use. Initially they claim that the data is a two-year long trace of IPTV in a large provider. Then they mention that they use a 9-days trace from 10 metropolitan areas, and then that they validate their model in a 2-weeks period. They have to elaborate more on their assumptions as well as the input data for the analysis and the validation. It is also important to verify that the 2-weeks validation period does not overlap with the 9 days they use to train the model.
- The mention that they use a 9-days trace from 10 metropolitan areas. Are all the metropolitan areas in the same time zone? Are there of the same population level? Does time zone or population have any effect on the model?

Response from the Authors

We would like to thank the reviewers for their insightful comments. We have taken careful consideration of all their comments. One reviewer has raised the point that more complex metrics, in addition to the VoD server bandwidth for a complete understanding of user behavior modeling. The computational overhead at the server is proportional to the number of stream control operations. To this end, we have provided in a companion paper [1], where a we did a careful micro- benchmarking of a VoD server by loading it with requests performing various stream control operations. We found that while the overheads are somewhat dependent on the type of operation, for the most part it is important to characterize correctly the number of operations, which we have done in this paper. With regards to aborted sessions and client interruption/rebuffering, our simulator faithfully captures the number of sessions that are aborted and the metric of client starvation. We note that this is very dependent on client implementation. We have added a plot on client starvation in the revised paper to address this comment.

Another suggestion from reviewers was to clarify the various data sets that we have used. We have used multiple data sets spanning over two years and a very large subscriber footprint. Specifically, our data sets include: 1) Data Set 1 (DS1) – for 120 days, from 9

VHOs, 3 Million STBs, from Jan/Apr/July/Nov 2009-2010, which we have used for the arrival process modeling; 2) Data Set 2 (DS2) – over 9 days, from 10 VHOs, 300K users, Jan. 9-17, 2010 used for characterization of the stream control FSM; 3) Data Set 3 (DS3) – for 60 days, from 10 VHOs, 300K users, in 2009, for studying the evolution of popularity. We have included a table in the paper to make this clear.

Finally, in response to the reviewer's request to better understand the independence between the training data and validation data, we note that in all cases the training and validation data were separate and independent. For modeling the arrival process of a particular day (e.g., Saturday), we pooled data from 16 Saturdays across a 2 year period (DS1) and validation is based on a randomly chosen weekday and weekend day from DS2. The DS1 and DS2 datasets are also from different days. Similarly, the finite state machine model was also trained and validated using different data sets.

[1] Pat Diminico, Vijay Gopalakrishnan, Rittwik Jana, Kadangode Ramakrishnan, Deborah Swayne, Vinay Vaishampayan, "Capacity Requirements for On-Demand IPTV Services", in Proc. of the third International Conference on COMmunication Systems and NETworkS (COMSNETS 2011), Bangalore, 2011.