Profiling Podcast-Based Content Distribution

Anirban Banerjee, Michalis Faloutsos and Laxmi N. Bhuyan
Department of Computer Science and Engineering
University of California, Riverside.
Email: anirban, michalis, bhuyan@cs.ucr.edu

***Abstract*— Media distribution via podcasts is a relatively new phenomenon which follows a different paradigm compared to traditional modes of content delivery. Despite its rapidly increasing subscriber base, podcast distribution has not been measured or modeled adequately, if at all. As our contribution, we develop a measurement based profile of podcasters. This profile consists of a comprehensive and detailed analysis of interesting characteristics of podcast streams which could be used for analytical studies and simulations. We start by conducting extensive active measurements to characterize 875 popular podcast streams for over a month. The take away message from our study is that podcast traffic is significantly different from the other types of traffic such as web traffic. For example, we find that podcast file sizes (between 2 and 35 MB) are not only significantly larger than web files on the average, but they follow a different distribution (a bimodal Gaussian compared to a heavy tail Pareto distribution for web files). Other interesting aspects of the podcast profile is the expected daily content download (per podcaster), in the range of 2 to 6 MB, and their content distribution patterns. We also find and quantify the heterogeneity in the intensity of content creation, since approximately 14% of podcasters contribute over 54% of files, amounting to about 30% of total byte-content.**

I. Introduction

Podcasts are a push-based mechanism for distributing multimedia files such as audio programs or music videos over the Internet. Podcast establishes streams (a.k.a. *feeds*) using either the RSS 2.0 or Atom syndication formats [11] and delivers content for playback on mobile devices and personal computers. The host or author of a podcast is called a podcaster. Podcast enabled web sites may offer direct download or streaming of their content. These content streams are distinguished by their ability to be downloaded automatically using software capable of reading RSS or Atom feeds.

Podcasting is already an important Internet application with roughly 6 million subscribers [5], and as such, it is an essential component of a complete model of the Interent traffic. Furthermore, podcasting is still growing rapidly towards a projected audience of 56 million by the year 2010 [8], [2], [3]. What started out as a system for distributing homespun radio programming over the Web has now caught on with big media companies. For example, ABC News, NBC News, ESPN, Disney, MTV, FOX, BBC, Apple, CNN and National Public Radio have all introduced podcast programming [8], [9], [10], [11]. Media retail services such as iTunes recently added 3,000 podcast programs to its iTunes online music store. In fact, one of the hubs for subscribing to podcasts, Feedburner.com, manages more podcasts than there are radio stations worldwide [7], and has been recently bought by Google. Further, to provide an idea of how much podcast content traverses the Internet everyday consider the following "conservative" back-of-the-envelope calculation. If 6 million users download an audio file of size 5 MB (a typical size as we see later) per day from only one podcaster, all this content-data amounts to a massive 30 TeraBytes. This number is indicative of the scale of podcast data being transferred and the popularity of this new technology. If we consider the more typical case where podcast listeners subscribe to multiple feeds, the total amount of podcast data can reach hundreds of TeraBytes.

Given its growing trend, we need to model the characteristics of podcasts, especially since podcast distribution differs from other content distribution applications. First, podcasting is a *push*-based distribution [10], [11], and thus it is different from the pull-based approach of web, real-time streaming, youtube-style video. Podcasting pivots on RSS enabled browsers and aggregators [8] which automatically download podcast content [12]. Prefetching of web content has some similarities to push-based approaches, but again it is ultimately user driven, based on popularity and not by *when* content is published by the content provider [26], [25],[27], [28]. Second, high volume websites and streaming video servers are generally hosted by carefully chosen servers, offered by specialized distribution companies like Akamai, with high-bandwith links. In contrast, popular podcast feeds are often home-grown and self-supported endeavors [14], and as such, podcast sources may not be hosted on high-speed servers or in "high-connectivity" network locations.

In this paper, we develop a measurement-based profile of podcasters, useful for developing traffic models of podcasting. To the best of our knowledge, this is the first extensive measurement study of podcasting. For example, this analysis could help answer network management and provisioning related questions and "what-if" scenarios given the growing trend of podcasting. The take away message from our study is that podcast traffic is significantly different from other types of traffic such as web traffic and thus needs to be analyzed separately.

We conduct active measurements, spanning a period of 30 days, from June to July 2006. We analyze 875 podcast streams [6], [7], [12] by using PlanetLab to enlist a diverse group of subscribers which connect to the selected podcasters and subsequently we log their performance. Our main contributions can be summarized in the following points.

a. A detailed profile of podcasting. Based on our measurements, we observe the following interesting characteris-

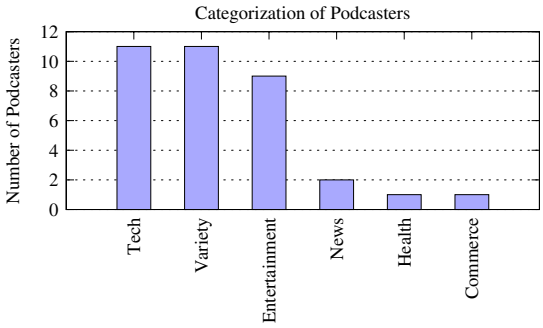


Fig. 1. Podcaster categories. Tech, variety and entertainment podcasters constitute the majority of podcasts.

tics of podcasting.

- **The podcast data profile is significantly different from web/http data:** The average podcast files is approximately 3 orders of magnitude larger than the the average http file. The average and median file sizes are 17 and 22 MB respectively for podcasts files compared to the average http file, which is less than 605KB according to three different studies [15], [16], [17]. In addition, podcast file sizes follow a different distribution, namely a skewed bimodal Gaussian distribution, compared to http files, which follow a heavy-tail Pareto distribution [18].
- **Content is not published uniformly throughout the day:** We observe that US based podcasters sparsely published content during 5AM to 12PM, US-Pacific Time (PST). Popular times for publishing content are 11 PM and 1 AM (PST).
- **Most podcasters publish new content every 5 to 16 hrs:** We observe that the time duration for podcasters publishing new content through respective feeds ranges from 5 to about 220 hrs. Most podcasters display intermission periods of about 5 to 16 hrs in-between publishing new content.
- **The expected content download per podcaster is 2 to 6 MB per day:** We find that a user can expect to download 2 to 6 MB of content per day from a podcaster. The average and median amount of content are 2.5 and 2 MB respectively.

b. A measurement-based traffic model. We synthesize our observations into an easy-to-interpret podcast traffic model. Our model provides both the qualitative (e.g. distributions of its behavior) and quantitative properties (ranges of values for each parameter), This traffic model can be used to generate synthetic podcast traffic which can be embedded into topology-graphs obtained from graph generators as GT-ITM [33].

The remainder of the paper, is organized as follows. Section II details our relevant literature, followed by Section III, concentrating on a data-centric analysis of podcast content. Section IV outlines a traffic model for podcasts, rounded up with an apposite conclusion.

II. Background

Podcasting is rapidly gaining large audiences [1]. Individuals with access to the Internet are able to publish and distribute podcasts without the need for resource-rich infrastructure. This is a significant deviation from prevalent commercial organizations which provide multimedia content using a subscription model, or employ high speed servers and fat-bandwidth links to disseminate content to end-users [2], [3], [4]. Most content is audio but can be video as well, in the form of news feeds, interview transcripts, entertainment and radio shows. One important research effort in this area describes a dynamic polling mechanism to reduce overhead incurred as a result of clients continuously polling content servers [30]. Our work differs significantly from this effort. We do not simulate end-user clients or propose a polling protocol. We focus on podcasts as a content delivery mechanism and quantify data and flow characteristics. Podcast data displays different characteristics when compared to content delivered by more traditional methods. Podcast data displays a different range for file sizes distributed to end-users compared to web/HTTP data. Research estimates report average page sizes for web pages to range from 60 to 605 KB [15], [16], [17]. This range is significantly different from podcast file size ranges by nearly an order of magnitude. Moreover, HTTP content displays a heavy tail Pareto distribution [18], different from podcast workload. Also, per-hour podcast traffic as observed from a client point of view follows a β distribution, unlike trends described for generic traffic in [32]. Also, real-time video and audio streaming is different from podcasting in terms of when data is transferred to end-users. Podcasting allows data to be disseminated, only when the content is published and hence data flows are bound by temporal characteristics of when content is published. Characterization of Autonomous Systems (ASs) based on their degree-based ranks has been described in [22] and we employ these methods in our research. Efforts as caching performance and workload characterization of document data [18], segment based caching, with blockwise variable sized segments [19], caching based on data migration protocols, and event-driven paradigms [20] and summary cache [21] mechanisms could all be used to improve content delivery for podcasts.

Statistical background: We now define some statistical distributions [29], which will be used in subsequent sections.

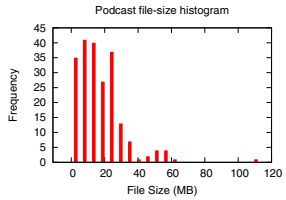
β **distribution**: Formally defined as:

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

where α and β both must be greater than zero, B defines the *Beta* function. This distribution is extensively employed in Bayesian statistics and is heavily used for PERT/critical-path-method based modeling.

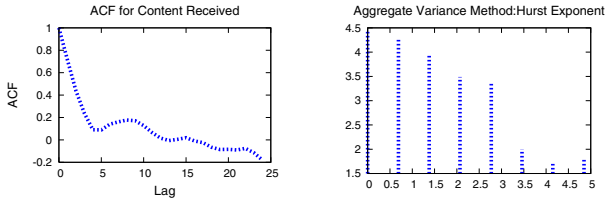
γ **distribution**: Formally defined by:

$$f(x;k,\theta) = x^{k-1} \frac{e^{\frac{-x}{\theta}}}{\theta^k \Gamma(k)}$$



(a) (b)

Fig. 2. (a) Histogram for podcast file sizes. (b) Bimodal Gaussian distribution, 95% confidence level.



(a) (b)

Fig. 3. (a)Autocorrelation Function for podcast content. ACF values upto lag=10 suggest the presence of memory in the system. Also, negative autocorrelation is observed after this range. (b)Hurst Exponent=0.681, Correlation Coefficient=96.16% (for ordered file sizes).

where k is the *shape parameter* and θ is the *scale parameter*, both greater than zero.

Bimodal distribution: A bimodal distribution is a distribution with two different peaks, with two distinct values that measurements tend to center around. Such distributions have been used to model population dynamics for groups of individuals.

III. Data Analysis

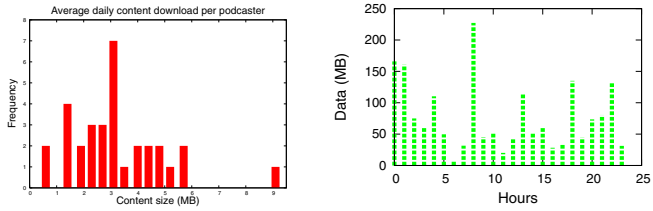
We begin with an explanation of the measurement setup.

A. Experimental Setup

We investigate 875 podcast streams from 35 podcasters (ranked according to number of subscribers) as listed on popular sources on the Internet [6], [8], [9], [10], [11], [12] and initiated connections for 30 days to each of these podcasters from PlanetLab clients. Subsequently, we logged traces of content being streamed from podcasters to the clients. Each podcast client located on PlanetLab nodes queried content servers every 20 minutes (similar to mean polling time mentioned in [30]) for new content. As soon as new content was detected, log files were updated to reflect temporal statistics. Content was downloaded to measure size and transfer latency. The majority of nodes were spread over the continental US (75%), while others were located in Europe (20%) and Asia (5%). To provide an idea of the kind of content being disseminated by these podcasters, we present Fig.1. Podcasters are classified by the various sites [6], [8], [9], [10], [11], [12] into technical, variety, entertainment, news, health and commerce categories. Podcasters in the technical category publish content related to hardware/software news and IT related events. Podcasters in variety category publish content related to current events, family radio shows, lifestyle while those in entertainment category publish music shows and Internet-radio programs. News podcasters publish current events, news reports and sports while those in commerce categories deal with management, investments and shares. Health related podcasters concentrate on general well being.

B. Data Analysis

1) **Podcast data profile:** We first analyze the characteristics of podcast data files. **Observation**: *Podcast content is different from http content*. There are two aspects to this: (a) The type of distribution followed by the file sizes and (b) the average value of the file-sizes.



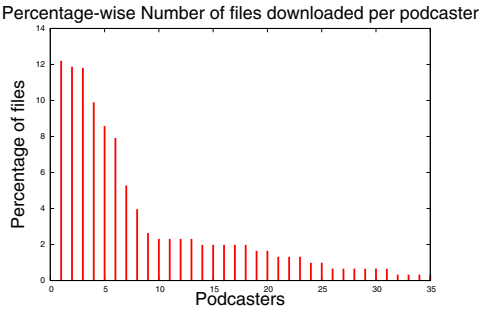
(a) (b)

Fig. 4. (a) Average content download per day per podcaster. (b) Hour-wise Content downloaded from podcasters, over the complete 30 day period.

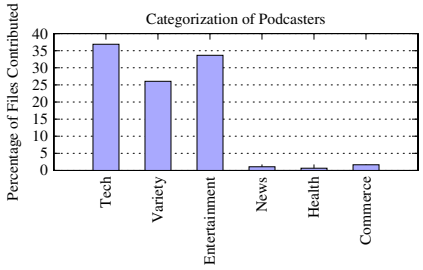
We present our findings in Fig.2.(a) where we plot file size in MB (X axis) versus frequency of files (Y axis) to show the distribution of individual files downloaded from podcasters. We observe that content size downloaded from all podcasters over the complete observation period ranges from 2 to 110 MB. However, 90.6% of files lie within a comparatively smaller range from 2 to 35 MB. *This observation clearly demarcates podcast content from web/http content* since the most probable sizes for podcast data is nearly an order of magnitude larger than average web/http content, about 60 to 605 KB [15], [16], [17], [18]. This is incorporated in SimPod.

Moreover, we observe that the distribution of file sizes for podcasts conforms to a bimodal Gaussian distribution whose PDF is displayed in Fig. 2.(b). The two Gaussian distributions can be defined by μ=13.5;σ^2=22 and μ=28;σ^2=50. The second distribution contributing the secondary mode observed in the form of a small hump as seen in Fig.2.(a). Fig. 2.(b) depicts a random sampling of values from a 0 to 1 range from these distributions based on a threshold probability of the bimodal distribution. We observe that for a threshold probability of 0.7, indicating that if a random sample has a lower magnitude, f(x, y)=N(μ=13.5,σ^2=22), else f(x, y)=N(μ=28,σ^2=50), the graph models the decay characteristics of the measured file sizes with less than 5% error. In contrast, file sizes for web/http objects are found to display a heavy tail Pareto distribution [18] which is different from the bimodal Gaussian distribution of the podcast data.

To verify that a unimodal distribution does not effectively model the file-size characteristics we compared the bimodal Gaussian distribution with a pure unimodal γ distribution and find that error rates for unimodal γ are 55.55% worse off than a bimodal Gaussian distribution.



(a)



(b)

Fig. 5. (a) Percentage of number of files contributed by each podcaster with respect to all files downloaded over a 30 day period. 14% of podcasters contribute nearly 54% of files. (b) The percentage of files as contributed by each category of podcasters.

Additionally, we attempt to quantify how much *memory* is present in the file arrival process, i.e., given a particular file size can we predict if the next few files received by the client will be of similar sizes? We present Fig.3.(a) which displays the auto-correlation function for the file sizes which are ordered in the manner they were received by clients. Each file is treated as a single sample point. We observe that ACF values upto 10 lags (files) indicate the presence of memory in the file arrival process. Beyond this range we observe negative correlation. We also test for long range dependence in the file arrival process. We present Fig.3b, which displays the Hurst parameter (H). It is found to be 0.681, which implies that the file arrival process exhibits long range dependence characteristics. These features are important for modeling purposes.

Observation: *A typical podcaster generates 2 to 6 MB of content per day*. Fig. 4.(a), where the X axis depicts content size (MB) versus frequency (Y axis), displays this fact. With podcasting set to garner larger audiences, this metric is significant for ISPs, who want to predict resource demand. Furthermore, end-users can allocate sufficient resources on personal machines to handle daily content downloads. Next, we present Fig.4.(b) which shows the total amount of content (over the complete 30 day period) downloaded by a client on an hourly basis. The spikes in the figure point to a large amount of content received during that hour. This data was found to conform to a β distribution with parameters, 2.0 and 24.2.

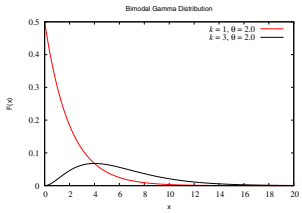
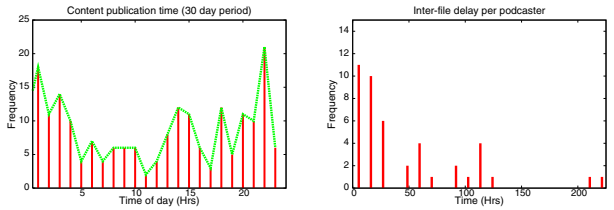


Fig. 6. Bimodal γ distribution which successfully reproduces the characteristics heterogeneity amongst podcasters for percentage of files contributed. The combined distribution can described by γ(k=1, θ=2) and γ(k=3, θ=2). 95% confidence level.

2) **Heterogeneity in podcaster activity:** We study the variance in the level of publication activity among different podcasters. We present Fig. 5.(a), depicting the percentage of total number of files (Y axis) each podcaster generates during our experiments. The X axis depicts the number of podcasters ranked by the number of files they generate. We observe a skewed distribution: 14% of podcasters contribute over 54% of files, which translates to about 30% of the byte-content. This indicates that a fraction of podcasters are responsible for the majority of content being disseminated. This is expected since certain podcasters host content which is published every few hours while others may not host content or shows which are disseminated as frequently. We find that a bimodal γ distribution, with γ(k=1, θ=2) and γ(k=3, θ=2), models the activity of podcasters, as displayed in Fig.6. To observe this phenomenon from a coarser granularity, we present Fig.5.(b). Clearly tech, variety and entertainment content providers supply the bulk of podcast data received.

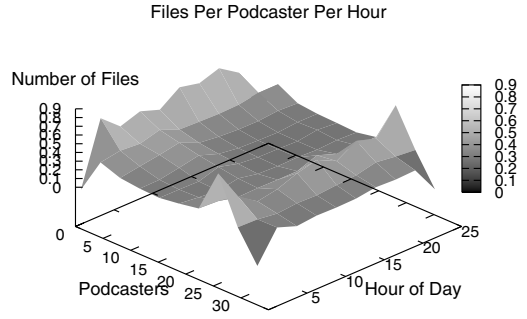
C. Analyzing temporal characteristics

Observation: *Podcast content is published sparsely between 5AM to 12 PM (US-PST)*. By performing a temporal analysis of podcast data generation, we ascertain when podcast content is published by podcasters. In Fig. 7.(a), X axis depicting the time of day (based on US-PST) and Y axis the frequency of publication of content. We see a timeline for podcast content publication. We observe relatively sedate activity between 5 AM to 12 PM for US based podcasters. This period crudely corresponds to office-hour time on the US east coast. Content is published during other periods of the day although not uniformly. Two clear peaks of publication activity are observed around 11 PM and 1 AM. Also, 3 AM, 2 to 3 PM and 6 PM, seem to be popular times for publication of content. Recall that these observations are averaged over a 30 day observation period. This possibly implies that podcast data is usually published during night hours for dissemination to audiences during the subsequent hours in the morning. Furthermore, we quantify the delay for publication of new content by a podcaster, which we will refer to as **inter-file** delay, in Fig. 7.(b). Again the X axis depicts time in hours, and Y axis the frequency. Clearly, a 5 to 16 hour inter-file delay period seems to be most prevalent. Also, approximately 58 and 112 hour inter-file delay also seem to be common. 112 and 58 hour inter-file delays could possibly correspond to shows that are broadcast once or

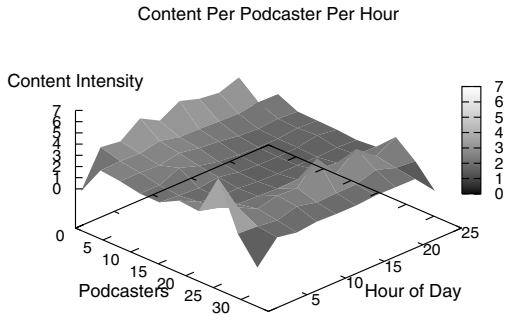


(a) (b)

Fig. 7. (a) Timeline for podcast content publication by podcasters, on a 24 hour scale. (b) Inter-file delay per podcaster, in hours. The most common inter-file delay ranges from about 2 to 16 hours.



(a)



(b)

Fig. 8. (a) Number of files sent per-podcaster, per-hour. X axis depicts the 35 podcasters, the Y axis depicts 0-24 hour timescale, while the Z axis depicts the normalized number of files sent by each podcaster during that time slot(b) Amount of content (Bytes) sent per-podcaster (normalized), per hour. Similar definitions hold true for axes.

thrice a week respectively. These metrics are important to understand the nature of podcast flows. Information such as a 5 hour inter-file delay can help ISPs understand the impact this kind of traffic as it passes through their networks.

We present a different view of the temporal analysis of podcasters in Fig.8. In Fig.8.(a) we observe files received from each podcaster in every one hour slot. In Fig.8.(b) we observe content received from each podcaster in the same one hour slot. From Fig.8.(a) we observe two consistent peaks running through the 24 hour spectrum. This provides insight regarding heterogeneity of podcasters. Further, in Fig.8.(b) we observe similar peaks running through the 24 hour spectrum again.

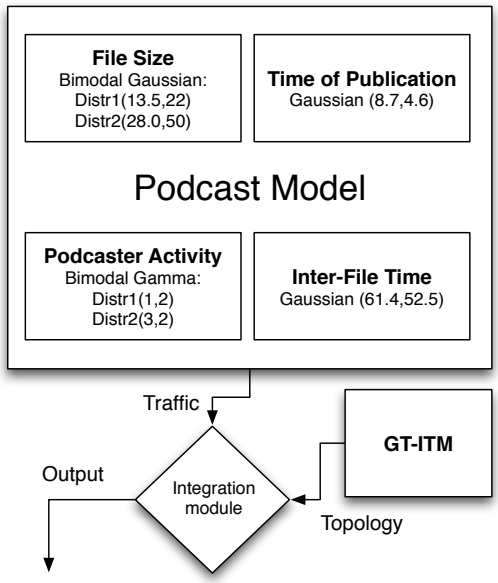


Fig. 9. Model for Measurement-based Podcast Traffic.

IV. Developing a Model for Podcasts

In previous sections, we have highlighted characteristics of podcast data. Primarily how podcasters differ from more traditional http content providers and this gives rise to the need for modeling this mode of content distribution. Podcasts are different not only in terms of content size from web data, but also with respect to temporal aspects. We use our analysis in previous sections to develop a traffic model. We begin by categorizing important features of podcasts into two base classes: **data** and **file generation**. Within each class we describe how to develop a model, which displays behavior similar to our data traces. In Fig. 9, we provide an overview of our model.

A. Data

This class defines the workload characteristics of a podcast model. File sizes follow a bimodal Gaussian distribution with parameters (13.5, 22) and (28, 50). This information encompasses deviation between file sizes for realism. To model unequal behavior of podcasters, the podcaster relative-activity metric, which can be easily implemented as a single bit allowing a particular podcaster to publish a file in an event driven simulation can be drawn from a bimodal γ mixture with shapes 1,2 and 3,2 respectively. Now, we address the time of publication and find that it should be drawn from a Gaussian distribution with parameters 8.7 and 4.6. Further, we can model the inter-file intervals between publishing content according to a Gaussian distribution with parameters 61.14 and 52.5.

B. File generation process

Here we present the analysis of the podcast file generation process. We test the measured file arrival process for stationarity in order to estimate parameters for synthetic modeling.

TABLE I

COEFFICIENTS AND RESIDUALS FOR THE ARMA MODELING OF THE FILE GENERATION PROCESS: PODCASTER TO CLIENT.

Min	$1Q$	$Median$	$3Q$	Max
-22.4528	-1.6312	-0.7010	0.7498	26.9933
Coefficients	Estimate	Std. Error	t value	Pr($>\|t\|$)
ar1	0.60996	0.06603	9.238	<2e-16
ma1	0.22086	0.08168	2.704	0.00685
intercept	2.62598	0.64550	4.068	4.74e05

This is imperative for successful Auto Regressive Moving Average [29] modeling. Applying the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [29], we can uncover if the incoming file process is level stationary or not. We find the p value for the test to be 0.1 with KPSS level at 0.273, implying that the stationarity hypothesis is true. To further substantiate this observation, we apply the augmented Dickey-Fuller test [29] on the incoming file process to confirm stationarity. We find the p value to be 0.01 with the Dickey-Fuller level at -4.357. This bolsters the claim that the file arrival process as seen by end-users is stationary. Now, we provide details of the ARMA (1,1) model to describe the file arrival process in Table. I. An ARMA (1,1) model is formally described in the following manner: $y_t = a_0 + a_1 y_{t-1} + b_1 e_{t-1}$. Where y_t represents the numeric vector or time series to be fit into the ARMA model and a_0 represents the intercept, while a_1 and b_1 are the estimated coefficients. The error variable is represented by the e_{t-x} series. The first two rows of Table. I define the statistics of the file size data received by a client. The following four rows depict estimates for the coefficients. The last column displays the "significance level" of each coefficient, all being below 0.05 which proves the efficacy of the model. Other similar models such as ARMA (1,2), ARMA(2,1) and ARMA(2,2) were not found to produce statistically significant estimates of coefficients.

This model could be used to generate podcast traffic patterns. This traffic could be assigned to network graphs created by topology generators like GT-ITM to provide a complete simulation environment.

V. CONCLUSION

This work conducts what is arguable the first extensive profiling effort of podcast traffic over 30 days.

A major conlusion of this work is that podcast content is inherently different from http/web content. The average podcast files is approximately 3 orders of magnitude larger than the the average http file: it is a difference between megabytes versus kilobytes. The average and median file sizes are 17 and 22 MB respectively for podcasts files compared to the average http file, which is less than 605KB. Furthermore, podcast file sizes follow a different distribution, namely a skewed bimodal Gaussian distribution, compared to the heavy-tail Pareto distribution of http files.

Another interesting observation is that podcasters are "nocturnal": a lot of content is published "after-hours", namely, between 11 PM and 1 AM (PST) for US-based podcasters. In contrast, content is rarely published during the early morning working hours, 5AM to 12PM, (PST).

Finally, we distill our observations into an easy-to-use podcast traffic model. This model captures both qualitative (e.g. distributions of its behavior) and quantitative aspects (ranges of values for each parameter), and can be readily used to generate synthetic podcast traffic.

Given the rising popularity of podcasting, it is important to measure, model and simulate podcasting, since it is bound to play an increasingly important role in the future Internet.

REFERENCES

[1] http://news.com/
[2] http://www.tdgresearch.com/
[3] http://news.bbc.co.uk/1/hi/technology /4658995.stm
[4] http://www.paulcolligan.com/2006/06/20
[5] http://www.edisonresearch.com/home/archives/LATimes060405.pdf
[6] http://www.podcastalley.com/
[7] http://blogs.feedburner.com/feedburner/archives/00175.html
[8] http://www.podcast.net/
[9] http://www.ipodder.org/
[10] http://www.podcastingnews.com/archives/2005
[11] http://en.wikipedia.org/wiki/Podcast
[12] http://www.podcastdirectory.com/podcasts/
[13] http://images.apple.com/education/solutions /podcasting
[14] http://www.smbtrendwire.com/2007/
[15] http://www2.sims.berkeley.edu/research /projects/how-much-info-2003/internet.htm -wbsamp
[16] http://www.dlib.org/dlib/april03/ lavoie/04lavoie.html
[17] http://www.pantos.org/atw/35654.html
[18] M. Arlitt and C. Williamson, Internet Web Servers: Workload Characterization and Performance Implications, IEEE/ACM Transactions on Networking , Vol. 5, No. 5, Oct. 1997.
[19] Kun-Lung Wu, Philip S. Yu and Joel L. Wolf, Segment-Based Proxy Caching of Multimedia Streams, Procs. WWW10, '01.
[20] Jia Song, Segment-based proxy caching for distributed cooperative media content servers, SIGOPS Op.Sys. Rev, vol.39, 05.
[21] Li Fan, Pei Cao, Jussara Almeida and Andrei Broder, Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol, Procs of ACM SIGCOMM'98, pp. 254-265.
[22] A. Banerjee, A. Mitra and M. Faloutsos, Dude where's my peer, Globecom 2006.
[23] Thomas Erlebach, Alexander Hall, and Thomas Schank: Classifying Customer-Provider Relationships in the Internet. IASTED Intl. Conf.on Comm.'s and Comp. Networks (CCN 02), pages 538-545, Nov'02.
[24] http://www.caida.org
[25] V. N. Padmanabhan and J. C.Mogul, Using predictive prefetching to improve World Wide Web latency, Sigcomm 96.
[26] A. Bestavros and C. Cunha, Server-initiated document dissemination for the WWW, IEEE Data Eng. Bulletin, Sep. 1996.
[27] M. Crovella and P. Batford, The network effects of prefetching, Procs. of Infocom 1998.
[28] Jia Wang, A Survey of Web Caching Schemes for the Internet, ACM Computer Communication Review (CCR), Vol. 29, No. 5, October 1999.
[29] A. M. Gun, M. K. Gupta and B. Dasgupta, An outline of statistical theory, The World Press, 2003.
[30] V. Ramasubramanian, R. Peterson and Emin Gun Sirer, Corona: A High Performance Publish-Subscribe System for the World Wide Web, in Procs. of NSDI 2006.
[31] http://www.podtrac.com
[32] K. J. Christensen and N. J. Javagal, Prediction of future world wide web traffic characteristics for capacity planning, Int. J. of Netw. Manag., Vol. 7, No. 5, 1997.
[33] http://www.cc.gatech.edu/projects/gtitm/
[34] http://www.alexa.com/site/ds/top_500