

Request Generation for a Peer-based PVR

Jeremy Guebert
Dept. of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
jlg327@cs.usask.ca

Dwight Makaroff
Dept. of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
makaroff@cs.usask.ca

Ketan Mayer Patel
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC
kmp@cs.unc.edu

ABSTRACT

Capacity planning, as well as protocol design and storage system design for video distribution systems are dependent on an understanding of the anticipated workload. Previous studies have focused on aggregate statistics of video popularity over time, and only recently has work been done which analyzes user behaviour variability over time. We are interested in a video storage/distribution system that uses peer resources to help content providers distribute video content to nodes in a network over long periods of time. Such a system is intended to operate on local principles of utility associated with each peer. More than ever, a workload model based on long-term user behaviour is required. In particular, it is unclear how request patterns driven by local utility can be shaped to match expected large-scale aggregate request characteristics such as a Zipf-distribution for overall popularity. In this paper, we describe early work in developing a workload generator for a Peer-based PVR (Personal Video Recorder) system to demonstrate some of these challenges.

Categories and Subject Descriptors

C.4 [Computer Systems Organization]: Modeling techniques

General Terms

Experimentation, Performance

Keywords

simulation, caching, video distribution, P2P, request generator

1. INTRODUCTION

Research efforts over the past two decades to understand access patterns to large-scale media servers, media content distribution networks, and video-on-demand systems generally reflect several common fundamental insights about the

distribution of requests for media objects. Most notably, it is widely understood that media object popularity usually has a Zipf-like distribution [2, 3, 12]. While specific details about the shapes of these distributions are, of course, going to be application and content specific, we can draw a number of important general inferences. First, for the most popular items, the high frequency of requests results in a situation where on average there is always more than one concurrent request for the media object. Second, the vast majority of media objects exist in the “tail” of the distribution. These objects are, for the most part, equally popular (or more aptly, unpopular). At any given time the number of current requests for these objects is either one or zero.

In light of these general insights, a designer of any media distribution system should be most concerned with only the most popular items. This is because system design can be optimized to exploit the concurrency of several outstanding requests in order to increase system performance be that measured by response time, buffering delay, server load, or some other objective function. Over the years a number of different techniques have been developed for doing so including batching, patching [6], prefix caching, and pyramid broadcasting [7, 13] just to name a few. For the most part, past work on media distribution systems has not been concerned with developing models for the popularity of individual media objects and/or the media requests of individual users. It has been sufficient to understand that some set of media objects will receive the bulk of the requests without necessarily needing to understand exactly which media objects these are or how the set changes over time.

Decentralized peer-to-peer systems, however, are fundamentally different than media distribution systems for which there is a global objective function for performance. In particular, each participant optimizes local resources to maximize the utility of participating relative to an objective function that may be unique to each individual client. For any given peer, the utility of a particular media object may be unrelated to the popularity of that object globally. In fact, we can expect the utility of a particular media object to be primarily determined by local conditions at the peer such as whether or not the media object has already been “consumed” (i.e., viewed) at least once already, the length of time that the object has been held by the peer, and the presence (or absence) of related media objects such as prior or subsequent episodes of a series.

An important difference between the utility of a media object with respect to an individual in a peer-to-peer system and the utility of a media object with respect to the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOSSDAV'10, June 2–4, 2010, Amsterdam, The Netherlands.
Copyright 2010 ACM 978-1-4503-0043-8/10/06 ...\$10.00.

aggregate of all media requests across the system is that we can not expect a Zipf-like distribution of utility among the media objects held by an individual peer. In other words, we do not expect an individual to value one media object twice as much as another media object simply because one media object is twice as popular with all the other peers in the system. The very fact that a peer has chosen to obtain a media object at all is an allocation of resources (both bandwidth and storage) that reflects a utility that is essentially infinitely greater than for objects not held. While some of these media objects will in fact be globally popular, most will not. From the perspective of the system as a whole (i.e., the perspective of a centralized or globally-designed system), only the most popular items will matter. From the perspective of an individual peer within a decentralized system, however, the majority of local resources are generally spent on media objects that have relatively high local utility but relatively low global popularity. In other words, if we look at the problem from the bottom-up, the tail matters. If the system is deployed in a decentralized fashion, centralized policies are unenforceable, and untenable.

We have observed some video workloads where the majority of the requests and (presumably) the majority of the bytes transferred are from objects in the tail of the distribution. Thus, the long term ability for servers to store these unpopular objects requires immense storage capacity or distribution over many nodes. Careful allocation of this storage must be performed by some entity in the system.

Thus, in a peer-based system that is active over long periods of time, it is not sufficient to determine the aggregate popularity of media objects at specific time scales. Instead, it is equally important to understand which users are interested in which videos at which times, since any caching system that is decentralized needs to operate on the *current utility* of the objects in the local cache, and some notion of the *current state* of the cache of peers. A more sophisticated model of workload requests is necessary. In particular, we must be able to generate media requests at the level of individual users which collectively demonstrates aggregate behaviour consistent with measurements taken in previous studies of centralized systems while at the same time is consistent with our intuition about how people actually value and use media objects locally.

Our goals in this paper are to:

- Describe a generalized model of a peer-to-peer media distribution system in which independently managed digital video recorders (i.e., TiVo-like devices) are able to communicate and share media objects.
- Report on early efforts to construct a model for media requests driven by peer-specific utilities for individual media objects that collectively generate an aggregate Zipf-like distribution.

The main contribution of this paper is to make the case that it is important to understand how large-scale Zipf-like aggregate media request distributions arise from individual peer-assigned object utilities. Our early experience suggests that doing so is challenging and not trivial. We believe that the early models we have developed in this paper provide an important starting point for addressing these challenges.

The rest of this paper is organized as follows. Section 2 contains a description of a peer-to-peer system driven by local utility that serves as a driving application. In Section 3,

we provide a survey of related work in the area of caching infrastructures for video as well as video request modeling and analysis. The system model implemented in our simulations is covered in Section 4, with particular emphasis on a workload model and workload generator. The design and results of simulation experiments are given in Section 5. These results lead us to some analysis and validation of the results against data captured from a deployed data set from a single subnet (edge) access log of a centralized system. Finally, we present conclusions and suggestions for future work in Section 6.

2. PEER-TO-PEER PVR

In this section, we describe a peer-based shared Personal Video Recorder (PVR) system that represents the kind of peer-to-peer media distribution networks that requires a request model grounded in user-assigned utility for specific media objects. A key element of this system is that each peer chooses its own objective-function for maximizing utility with respect to local resources.

Our envisioned model is an extension of a PVR that is able to record content broadcast by a set of content-providers (i.e., channels on a digital cable or satellite system) for the purpose of viewing the material at a later time (i.e., time shifting) or repeatedly in the future (i.e., longer-term archiving). Such devices (e.g., TiVo) are widely deployed today and are generally included as a component of digital cable and digital satellite services.

Extending this PVR model with peer-to-peer communication allows peers to share the contents of their local PVR. By doing so, any media object can be requested by any peer at any time after the media object has been distributed originally. Obviously there are costs associated with both storing a local copy (i.e., space in the PVR) as well as with requesting a non-local copy from a peer (i.e., bandwidth). Thus, each individual peer needs to make decisions as to which media objects to cache upon broadcast and which can remain uncached. Currently, these decisions are purely local since peer-to-peer communication does not exist among PVR users. These decisions are reasonably straightforward since the only factor to be considered is whether the utility of a newly available media object is greater than the utility of some object already held in the PVR cache. If so, the new object is cached in place of an existing, less useful object.

A peer-to-peer PVR system, however, is complicated by inter-peer dynamics of utility. In particular, the cost of obtaining a popular media object are inversely proportional to its global popularity. In other words, a very popular object is easier to obtain because there are a greater number of peers who can be expected to have the object. Conversely, any media object that has a degree of local utility for a particular peer that is relatively unpopular globally among all peers is very costly to retrieve since many fewer peers also have a cached copy. We envision that the combination of these two competing forces will lead to interesting and complex strategies that govern how peers behave. For example, if too many peers rely on the expectation of other peers caching media objects that are globally popular, the result will be too few copies of the popular content with respect to the aggregate demand for that content. While this model of interaction is similar to how BitTorrent¹ clients interact, only recent en-

¹www.bittorrent.com

hancements to BitTorrent permit streaming, and our target system would also incorporate object utility and incentives to share in a comprehensive manner.

Furthermore, we can envision a number of interesting strategies that are enabled by adaptation techniques such as multiple description coding and forward error correction. Video data is unique in that it can be scaled/subdivided on many dimensions (temporal, spatial, colour), each of which can generate a different version of a particular quality. Depending on the encoding, different versions can be combined to provide a higher-quality version of the object (i.e. base and enhancement layers). If peers probabilistically cache different versions of each object, the quality of a video object can be increased by exchanging versions cached at other peers.

We want to provide a system without centralized controls, but with common policies, such that each user can cache content in order to maximize local utility for the objects cached. At the same time, however, high global utility is achieved by providing appropriate incentives for sharing. Such a model requires that the long-term behaviour of the users is modeled, since caching video content for use by peers implies that this cache will have a long lifetime relative to the viewing time of a single object. Furthermore, we expect users to have a non-zero utility associated with archiving objects with relatively low global popularity in order to maintain access to those objects. In other words, a peer should prefer to maintain a copy of a media object that would otherwise disappear from the system entirely.

To explore this interesting system space, however, requires the ability to generate peer-specific utility values for each media object. Unfortunately, prior studies of media access provide only a view of aggregate media access distributions. Furthermore, simply projecting the aggregate distribution to each peer is not reasonable with respect to how one would expect a real user to behave. For example, while the aggregate demand for each media object can be expected to maintain a Zipf-like distribution with respect to its overall popularity rank, one can not expect the requests generated by individual users to be Zipf-like at all. In other words, if in the aggregate distribution the most popular movie is expected to be twice as popular as the second most popular movie, we do not expect any particular user to consume the most popular movie twice as often as the second most popular movie. What we do expect is that most peers consume the most popular movie once (or a small number of times) and half as many will consume the second most popular movie once (or a small number of times), resulting in the observed aggregate effect.

The challenge that remains is to construct a model for generating peer-specific utilities for media objects that in aggregate create demand for media objects that mirror the Zipf-like distributions that we know exist, but that exhibit reasonable local characteristics. In particular, we would expect that local utility for a media object is coherent (i.e., does not change randomly with time) and that utility generally decays with both consumption and time.

3. RELATED WORK

Choong *et al.* describe a content distribution system that inserts a proxy layer which distributes rewards to clients for storing/forwarding media content within the scope of the local last-mile network [1]. Peer-to-peer streaming with selfish competition is introduced by Shrivastava and Banerjee [11].

They describe natural incentives in such a model to improve the quality of media streams by collaborating.

Huang *et al.* evaluate the PPLive VoD System in developing principles for the architecture of P2P VoD systems [8]. They consider similar issues to our model, such as video segmentation and replication strategy. They measure user behavior and user satisfaction on a small window of time (1 week) and carefully examine many short-term properties of user behaviour and object popularity.

The distribution of video requests has been studied for many years and has not been resolved to any degree of certainty in the existing literature. One of the first investigations was by Griwodz *et al.* [4], which predated deployments of on-line video distribution systems. The authors developed a model for popularity decay that was exponential and superimposed a random effect dependent on the size of the user population. They found that long-term trends differed from day-to-day measurements.

Jin and Bestavros developed the GISMO workload generator [9] to provide more realistic workload traces than had previously been used in media server analysis and design. Their work is more comprehensive in that the model generates demand requests that consider the inter-arrival time of sessions, size distributions and user-interactivity behaviour, in addition to the bit-rate requirement variability over time. Our model is significantly simpler in scope, and more specifically deals with a restricted mechanism for making new content available in the system.

Guo *et al.* [5] show that several internet media access patterns exhibit the characteristics of a stretched exponential distribution, and are not zipf-like, when only the media objects themselves are considered. Many data requests are for metadata or content associated with an ad server. In addition, much content is filtered by caches along the way, so that accurate measurements from either the client side or the server side are difficult. When these requests are removed, the video request patterns in 16 different data sets were analyzed and showed a good fit to the stretched exponential distribution.

Gill *et al.* found that YouTube video requests on a university campus do exhibit a Zipf distribution [3], and otherwise confirm similar results of other previous work. There are diurnal effects, and day of week effects visible in the data. News and sports have high, but short-term popularity, while music, comedy, and other entertainment videos are most popular in the long-term. Garcia *et al.* analyze the user interactions in a News-On-Demand website [2]. The authors divide interactions into sessions, requests and interactions. They analyze the interdependence of the various variables used as parameters for request generators.

Any peer-based system will exhibit characteristics of social networking. Ostrowski and Sarhan evaluated traces of social networking video [10]. They found Zipf distributions for the top videos over different time intervals, as well as diurnal and weekly viewing patterns.

An artifact of currently-deployed systems is that the capabilities of the technology significantly affect the behaviour of the users. In particular, slow network connections may result in an incomplete viewing or may cause the user to time-shift to a time when the network is less busy. At the extreme, users may choose to download, rather than stream the content. Users adapt to the limitations of the system, and as such, the requests captured in a trace may not reflect

their desired watching behaviour. These traces do, however, provide a potential target request pattern distributions that our system should be able to generate. Since the evidence from measurement studies is sufficiently contradictory, our model should be capable of generating requests that ultimately match a wide array of popularity distributions.

4. SYSTEM/SIMULATION MODEL

The architecture of our model is based on two entities in the system: a client/peer node and a content server. They store information about two other entities: a user and a video. Peers and the server process *viewing requests* and execute *cache management operations*. The size and length of the simulation is governed by several parameters, which are factors in the performance/scalability experiments performed in the next section.

The parameters of size include: a) Number of channels, b) Number of videos and c) Number of users. The parameter of configuration is the decay factor for utility in time.

The *content provider* publishes a number of new videos every time period equal to the number of channels. The initial popularity of each video is determined by the system to correspond to a Zipf distribution with particular parameter α . From this initial popularity, probabilistic peer-specific utilities are derived. During each time period, a user generates a request for a video chosen by a probabilistic process weighted by its peer-specific utility values.

Every time unit, utility values of each user for each video previously published are adjusted. For simplicity, we conjecture that popularity decays with time since release for normal videos, so, all existing videos are decayed. Furthermore, once a user has viewed a video, the utility of that video is discounted. Thus, each viewed video in the simulation has an extra decay applied for the users that watched that video in the current time period.

Because of the time varying nature of a video's popularity, and the current user's viewing behaviour, the utility for an object will change over time. The policy to be followed at the client may be complicated, but its basic nature will be to retain the most valuable objects in the local cache, so as to maximize local utility. This will require the client to perform one or more of the following activities: a) purge less desirable content, b) transcode content that is still valuable, but less valuable than in the previous time period, and c) pre-fetch content for which the utility has increased enough to displace previously cached content.

5. EXPERIMENTS

The simulator generates requests by a static user population from a video population that changes over time. Videos are added, but not removed from the population each publishing period. A run of the simulator consists of publishing the specified number of videos on the specified number of channels, and issuing requests for available videos at each publishing time. The number of time units that the simulation runs is thus the ratio of videos to channels. We selected the Zipf distribution as the target distribution, because it matches most of the previous research. Future work will examine request generators that can produce requests according to other distributions.

The objective of the experiments is to generate a representative sample of video requests from the population of users

and videos and determine whether the original Zipf distribution of popularity used to derive peer-specific utilities is recreated by the aggregate of peer requests across time. The video requests should match a target distribution in terms of aggregate popularity at multiple time scales. As well, it is desirable that the component behaviour of an individual user and the requests for individual videos conform to our intuition regarding the proposed interaction model for a user and the system, both as instantaneous measurements and over long periods of time. In particular, it is not clear whether or not such aggregate behavior properly emerges given utility decay and the introduction of new video objects.

Preliminary experiments varied the number of users, the number of videos, and the number of channels in an independent fashion with a constant decay factor. The next round of experiments kept the size parameters constant and modified the decay parameter. This was in an effort to determine how to control the time-varying nature of object popularity so that the overall distribution remained the same over all time periods.

We performed a number of experiments that varied the number of users, the number of videos and the number of channels. As expected, the major results of these experiments were a change in the density of traffic, but not in the shape of the popularity of the objects relative to each other. After generating many experimental runs, we graphed the popularity of a video object against its rank within the set of all objects. When graphed on a log-log scale, data that exhibits a Zipf distribution results in a straight line with a negative slope, indicating the Zipf parameter of the distribution. In the following graphs, the lines represent the average of 10 replications of a request generating session with the same configuration. The deviation is very small (less than 0.03%), so error bars are not shown.

Figure 1 shows the results for one sample experiment. Except for the very tail of the graph, the video requests form a Zipf distribution, since decay was chosen appropriately.

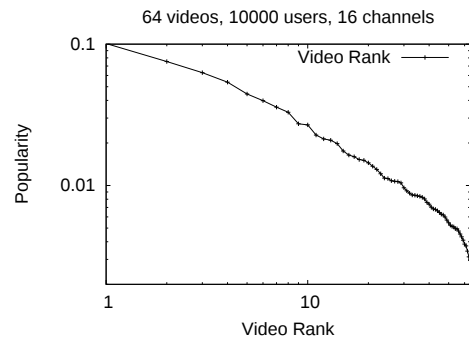


Figure 1: Sample Rank vs. Popularity

Figure 2 shows two runs that have 5000 users, and 128 videos with 0.29 for the decay value. The simulation lasts for more time units in one run than the other, because of the different numbers of channels. When 64 channels are utilized, the simulation lasts 2 time units, whereas when we publish on 16 channels, the simulation lasts 8 time units. We can see that the higher popularity videos (the top 6) in

the 16 channel case do not show a Zipf distribution, because they decay too quickly; most of the requests for an object come in the initial publishing time interval. We also see a drop off in the tail of the distribution for the 64 channel case, possibly because second interval low popularity videos have only one time interval in which to be selected.

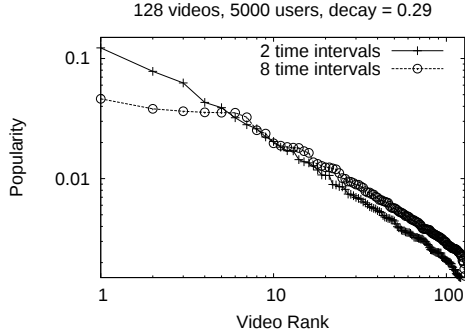


Figure 2: Effect of Number of Channels (64 and 16)

The effect of an improper decay value can be seen another way. In Figure 3, what can be seen is that the tail of the distribution drops off if the decay factor is incorrect. If

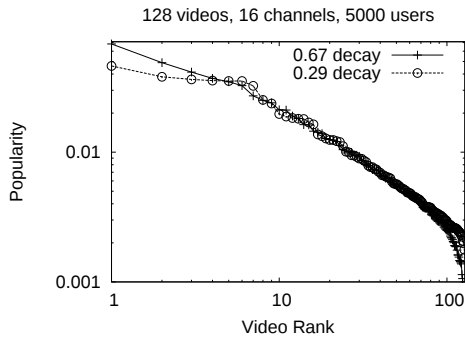


Figure 3: Effect of Decay (8 time units)

the decay factor is too small (i.e. decay occurs too slowly), then the older videos continue to be very popular and newly released videos that are popular within the set of videos that were released at the same time never become globally popular. This is inconsistent with our experience with real-world systems. If decay is too large, then videos only get watched in the first few time periods after release and in the long term, there will be a large number of videos with approximately the same popularity, namely those that were tagged as the most popular at publishing/release time. Future work is required to determine a more realistic pattern of decay for accurate aggregate behaviour for videos in the tail of the distribution.

Figure 4 shows video requests over time for one of the most popular video objects for a run that spanned 100 time units with 5000 users and 10 channels. Only the first 30

units of time are shown as very few requests occurred later. Similar results were obtained for other popular videos.

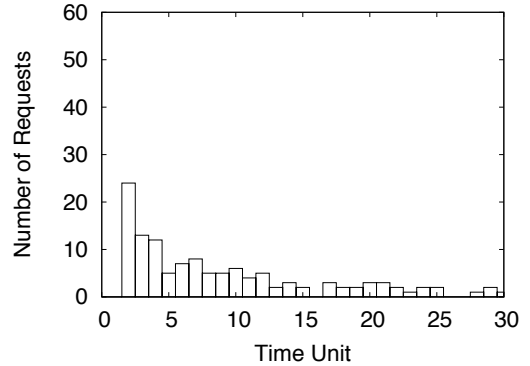


Figure 4: Third Most Popular Video (Model)

To determine if our simulation data is qualitatively similar to observed behaviour in a deployed system, we have obtained an anonymized data set of YouTube video requests from the University of Calgary (obtained by permission) [3]. The identity of the requester was made anonymous, but the identity of the video object was retained. Gill *et al.* performed an exhaustive analysis of the characteristics of this data set from the request level to the byte level, and over different time-scales. This is a somewhat similar in context to our envisioned deployment scenario, but we are only interested in the requests for video objects over time by different users. Nevertheless, real data from a deployed system provides us with a large comparison data set.

We expected that object popularity would decline as a function of time since release. Some of the videos showed this property, but some did not. The most popular video is requested at the beginning of the trace period, reaches its popularity peak at day 30, and then shows a decline, but still has a steady number of requests until the end of the trace period. We show the third most popular video (Figure 5) as an example of the decay pattern observed in 5 of the top 10 videos from the YouTube data set.

This compares well to our results from the simulator. It is also appropriate to note that videos released late in the capture time period were able to become globally as popular over the time period as those released earlier. The third most popular video was not viewed at all until Day 60. The request distribution over time for other objects shows very different decay patterns, indicating that several models will be required in a full-featured request generator.

An analysis of the tail of the distribution showed that over three months of data collection 99% of the objects were requested fewer than 8 times. As well, 90% of the requests were made to these objects in the long-tail. The relative size of each object was not analyzed, but if objects are close to the same size, then a significant majority of the bytes transferred are also contained in these requests.

The user-based approach is necessary to provide the basis for a user having utility in keeping certain content in their long-term cache. First of all, it is clear that we can model individual requests for videos in a manner consis-

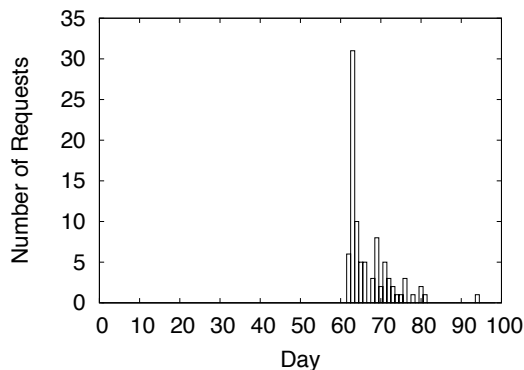


Figure 5: Third Most Popular Video (YouTube)

tent with some aspects of at least one case of real-world data. Users have a time-varying behaviour that influences their own personal storage requirements. Secondly, the time varying nature of the requests for an individual video also shows similarity to some video objects in the case study. This is encouraging, but is also an area for further work in which alternate video types are incorporated that show a different decay pattern. These different decay patterns may influence other factors in the system, such as desired level of replication of a particular object. Finally, we note that the aggregate request behaviour for all objects still follows a zipf distribution if we start each distribution period with a set of videos explicitly chosen from a zipf distribution and decay values are carefully chosen. Further work will verify if these properties can be maintained with other distributions and if critical decay values can be analytically derived.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we make a case for developing a user-specific utility-based request model for media objects. We believe having such a model will be important for developing and evaluation decentralized peer-to-peer media distribution systems. In particular, we anticipate that any such model will need to incorporate realistic user behavior such as utility decay over time and due to consumption, the introduction of new media objects over time, and aggregate request distributions that are Zipf-like and thus consistent with prior studies of centralized systems.

We implemented a request generator that utilized global popularities of videos, associating each request with a user. The results showed that we can artificially generate a pattern of requests that has a Zipf distribution by selection of appropriate decay parameters and arbitrarily assigning initial popularity values to each video. However, the resulting distribution is sensitive to the specific decay values used and the scale of the simulation. While request patterns for specific media objects compared favorably with patterns seen in empirical measurements, the empirical data exhibit a number of patterns that our model is unable to generate.

Future work is required to develop more accurate models of request arrivals, distribution of requests over users and request patterns from individual users for individual videos.

In particular, we intend to make the workload model more realistic by considering viewing operations taking place in between publishing periods and extending the specifications of media object types (length, genre). A dynamic population of users would also be more realistic.

Acknowledgements

The authors would like to thank the Networks Research Research Group at the University of Calgary for use of the anonymized YouTube data set.

7. REFERENCES

- [1] K. N. Choong, B. M. Ali, V. Prakash, and Y. C. Yee. The Case for a Dynamic Proxy Framework with Voluntary/Incentive-based Client Hosting Scheme. *Journal of Network and Computer Applications*, 30:167–185, 2007.
- [2] R. García, X. G. Pañeda, D. Melendi, and V. Garcia. Probabilistic Analysis and Interdependence Discovery in the User Interactions of a Video News on Demand Service. *Computer Networks*, 53(12):2038–2049, 2009.
- [3] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A View From The Edge. In *IMC*, pages 15–28, San Diego, CA, October 2007.
- [4] C. Griwodz, M. Bar, and L. Wolf. Long-term Movie Popularity Models in Video-on-Demand Systems or the Life of an on-Demand Movie. In *ACM Multimedia*, pages 349–357, Seattle, WA, October 1997.
- [5] L. Guo, E. Tan, S. Chen, Z. Ziao, and X. Zhang. The Stretched Exponential Distribution of Internet Media Access Patterns. In *PODC*, pages 283–294, Toronto, ON, Canada, August 2008.
- [6] K. Hua, Y. Chai, and S. Sheu. Patching: A Multicast Technique for True Video-on-Demand Services. In *ACM Multimedia*, pages 191–200, September 1998.
- [7] K. Hua and S. Sheu. Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-On-Demand Systems. In *ACM SIGCOMM*, pages 89–100, September 1997.
- [8] Y. Huang, T. Fu, D. Ching, J. Lui, and C. Huang. Challenges, Design and Analysis of a Large-scale P2P-VOD system. In *ACM SIGCOMM*, pages 375–388, Seattle, WA, USA, August 2008.
- [9] S. Jin and A. Bestavros. Gismo: a Generator of Internet Streaming Media Objects and Workloads. *SIGMETRICS Perform. Eval. Rev.*, 29(3):2–10, 2001.
- [10] J. Ostrowski and N. Sarhan. Characterization of Social Video. In *MMCN*, pages 347–356, San Jose, CA, January 2009.
- [11] V. Shrivastava and S. Banerjee. Natural Selection in Peer-to-Peer Streaming: From the Cathedral to the Bazaar. In *NOSSDAV '05*, pages 93–98, Stevenson, WA, June 2005.
- [12] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat. Long-term Streaming Media Server Workload Analysis and Modeling. Technical Report HPL-2003-23, HP Labs, Palo Alto, CA, January 2003.
- [13] S. Viswanathan and T. Imielinski. Metropolitan Area Video-On-Demand Service Using Pyramid Broadcasting. *Multimedia Systems*, 4:197–208, 1996.