

Can Internet Video-on-Demand be Profitable?

Cheng Huang
Microsoft Research
Redmond, WA 98052

Jin Li
Microsoft Research
Redmond, WA 98052

Keith W. Ross
Polytechnic University
Brooklyn, NY 11201

ABSTRACT

Video-on-demand in the Internet has become an immensely popular service in recent years. But due to its high bandwidth requirements and popularity, it is also a costly service to provide. We consider the design and potential benefits of peer-assisted video-on-demand, in which participating peers assist the server in delivering VoD content. The assistance is done in such a way that it provides the same user quality experience as pure client-server distribution. We focus on the single-video approach, whereby a peer only redistributes a video that it is currently watching.

Using a nine-month trace from a client-server VoD deployment for MSN Video, we assess what the 95 percentile server bandwidth costs would have been if a peer-assisted employment had been instead used. We show that peer-assistance can dramatically reduce server bandwidth costs, particularly if peers prefetch content when there is spare upload capacity in the system. We consider the impact of peer-assisted VoD on the cross-traffic among ISPs. Although this traffic is significant, if care is taken to localize the P2P traffic within the ISPs, we can eliminate the ISP cross traffic while still achieving important reductions in server bandwidth. We also develop a simple analytical model which captures many of the critical features of peer-assisted VoD, including its operational modes.

Categories and Subject Descriptors

C.2 [Computer-Communication Networks]: Distributed Systems

General Terms

Measurement, Performance

Keywords

peer-to-peer, video-on-demand, ISP-friendly

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.

SIGCOMM'07, August 27–31, 2007, Kyoto, Japan.
Copyright 2007 ACM 978-1-59593-713-1/07/0008 ...\$5.00.

1. INTRODUCTION

Video-on-demand (VoD, also called on-demand video streaming) has become an extremely popular service in the Internet. For example, YouTube, a video-sharing service which streams its videos to users on-demand, has about 20 million views a day with a total viewing time of over 10,000 years to date [1]. Other major Internet VoD publishers include MSN Video, Google Video, Yahoo Video, CNN, and a plethora of copycat YouTube sites.

Most of the VoD being streamed over the Internet today is encoded in the 200-400 kbps range. At these rates, ISPs (or CDNs) typically charge video publishers 0.1 to 1.0 cent per video minute. It has been estimated that YouTube pays over 1 million dollars a month in bandwidth costs. We show in this paper that the MSN VoD service has a billing rate corresponding to 2.20 Gbps (as of December 2006). These costs are expected to go up, as demand increases and higher-quality videos (with rates up to 3 Mbps or more) are made available.

Currently, none of the Internet VoD providers earn significant revenues from their services. Internet VoD providers will likely attempt to monetize their services in the near future using embedded video advertisements, subscriptions, or pay-per-views (or some combination thereof). But given the enormous costs associated with client-server distribution – due both to the increasing video quality and to the enormous demand – the revenues very possibly will not cover the cost of providing the service. In other words, although VoD has become one of the most popular Internet services today, the service is, and will likely continue to be, unprofitable with client-server distribution.

In this paper, we consider the design and potential benefits of *peer-assisted VoD*. In peer-assisted VoD, there is still a server (or server farm) which stores all of the publisher's videos and guarantees that users playback the video at the playback rate without any quality degradation. But in peer-assisted VoD, the peers that are viewing the publisher's videos also assist in redistributing the videos. Since peer-assisted VoD can move a significant fraction of the uploading from the server to the peers, it can potentially dramatically reduce the publisher's bandwidth costs.

There are two broad design approaches to peer-assisted VoD. In one approach, which we refer to as the *single video approach*, a peer only redistributes the video it is currently watching; it does not redistribute video that it may have watched and stored in the past. The single-video approach is similar to a torrent in BitTorrent [11,12], in which all peers in the torrent share exactly one file. In the second approach,

which we refer to as the *multiple video approach*, a peer can redistribute a video that it previously viewed but is currently not viewing. Compared with the multiple-video approach, the single-video approach is simpler in the client and tracker design, and involves a straightforward end-user policy. Our focus in this paper is on the single-video approach, which intuitively should provide good performance if a large fraction of the requests are for a relatively small number of the publisher’s videos. If a publisher has N videos and uses the single-video approach, then the distribution problem essentially becomes N sub-distribution problems, one for each video.

In a peer-assisted VoD system, when the peers *alone* cannot redistribute the video among each other, the server makes up the difference, so that each peer receives the video at the (encoded) playback rate. The server is only active when the peers alone cannot satisfy the demand. When the peers alone can satisfy the demand, not only is the server inactive, but the peers can potentially prefetch video from each other using the peers’ surplus bandwidth. This prefetching allows the peers build a reservoir of video, which can be tapped when the aggregate upload bandwidth of peers becomes less than the demand across all peers. The contributions of this paper are as follows:

- We collect a nine-month trace from the MSN Video service. This data covers over 520 million streaming requests for more than 59,000 videos. During this period, MSN used a client-server deployment. We process this data to extract many of the key characteristics of a large-scale Internet VoD deployment. Particular attention is given to the characteristics that are relevant to a peer-assisted deployment. To our knowledge, this paper presents the first measurement study of an on-demand video streaming system in such a large scale.
- We present a simple theory for peer-assisted VoD. This theory identifies three basic operating modes of peer-assisted VoD system: the surplus mode, when the aggregate upload bandwidth supply of the peers exceeds the aggregate bandwidth demand; the balanced mode, when the bandwidth supply is in the vicinity of the bandwidth demand; and the deficit mode, when the bandwidth supply is less than the demand. The model captures many of the essential features of peer-assisted VoD. In particular, it shows that when a service provider is operating in the surplus mode, it can increase the quality of the videos without incurring significant additional server bandwidth costs (using peer-assisted distribution). However, when operating in the deficit mode, server bandwidth costs increase linearly with video bit rates. These economic considerations dictate that the natural mode of operation for peer-assisted VoD is in the balanced mode.
- For the single-video approach, we describe three natural prefetching policies for exploiting surplus peer upload capacity. These policies are *no-prefetching*, *water-leveling*, and *greedy*. We show that they can significantly reduce server bandwidth usage, with prefetching providing dramatic gains. Moreover, the prefetching policies perform close to a bound, and are therefore nearly optimal.
- We use the nine-month MSN trace, which was col-

lected for a client-server deployment, to drive simulations for peer-assisted deployments. Using this trace simulation, we closely examine the impact of early peer departures and user interactivity (e.g., pause/resume and skipping content). The results show that if peer-assisted distribution had been instead employed, significant reductions in bandwidth costs would have resulted. Furthermore, early departures and user interactivity do not substantially reduce these gains. For example, the client-server bandwidth requirement (using the 95 percentile rule) for the month of December 2006 was 2.20 Gbps; if peer-assisted distribution had been deployed, with the greedy prefetch policy, the server bandwidth requirement could have potentially been reduced to 79 Mbps.

- We also explore the impact of peer-assisted VoD on ISPs. Indeed, peer-assisted VoD shifts the uploading burden from the servers to the peers (as does BitTorrent). The P2P uploading in turn can generate significant cross-ISP traffic. We examine an ISP-friendly peer-assisted scheme, which localizes the P2P traffic within ISPs (and hence generates no cross-ISP traffic). Using the trace data, we show that there are still significant server bandwidth cost reductions when the peer-assisted VoD scheme is ISP friendly.

This paper is organized as follows. In Section 2 we describe the nine-month MSN trace and report on observations about the video popularity distribution, user demand and upload resources, user interactivity and service evolution. In Section 3 we present a theory for peer-assisted VoD, which exposes many of the fundamental issues in peer-assisted VoD. In Section 4, we use the MSN trace data to drive simulations of peer-assisted distribution schemes. Here we examine the impact of user interactivity, the possible gains with peer-assistance and prefetching, total costs using the 95 percentile rule, and the impact of the most and least popular videos on the total cost. In Section 5 we examine the costs that peer-assisted VoD place on local ISPs and explore how these costs can be minimized. In Section 6 we describe related work, and finally in Section 7 we conclude.

2. CHARACTERISTICS OF A LARGE SCALE VOD SERVICE

We collected data from the popular MSN Video site for a nine-month period, from April through December 2006. The raw data captures all the activities at the MSN video site, including live and on-demand streaming sessions for both audio and video. We filtered out the audio and live sessions, as the main focus of this study is on on-demand video streaming. The resulting filtered trace contains in total over 520 million streaming requests for more than 59,000 video files. All video streams were served from MSN servers (via a CDN) to clients using RTSP, Microsoft’s HTTP streaming protocol, or Media Server (MMS) protocol. (Peers did not assist in the streaming during this time.)

In this section, we report observations from this data set that shed insight on an eventual peer-assisted deployment for VoD. To our knowledge, this section presents the first measurement study of a large-scale on-demand video streaming system. In Sections 4-5, we will use the trace data to explore the design and potential benefits of peer-assisted VoD.

2.1 Trace Records

The MSN Video trace contains *trace records*, with each record characterizing a portion of a streaming session, as described below. Each trace record contains over 50 fields [2]. Using three field categories, we now summarize the fields that are of interest in this study:

1. *Client Information Fields*: All streaming clients are Windows Media Players (WMP). Each trace record contains the player ID, version, and language; the public IP address of the client host; the host web browser and version (the MSN Video service is accessible through its website); and the OS/version and CPU of the host machine.
2. *Video Content Fields*: Each trace record includes the video file name, the video file length (in seconds), and the size of the file. From the video length and the file size, we compute the average bitrate of each video.¹
3. *Streaming Fields*: Each trace record includes the time when the client establishes a connection to the server; where in the video the client starts playback; and how many seconds of data the client has played. It also includes an indication of whether the playback is real-time or fast-forward/backward. Note that although Windows Media Service allows for a “fast caching” mode, where data is streamed to clients faster than playback rate (e.g. twice as fast as playback rate), this mode is disabled in the MSN Video service. Hence, the length of the client playback is equivalent to the length of video streamed.

Each interactive operation (pause/resume, fast-forward, fast-backward, and repositioning) generates a new trace record. Hence, often there are multiple trace records from the same player within the same streaming session.

2.2 Identifying Users and Streaming Sessions

For our analysis, we would like to be able to identify the trace records that were generated by the same Windows Media Player (WMP) (that is, generated by the same user). As indicated above, each trace record includes a player ID, which in principle can be used to identify the players. However, in our nine-month trace, only 7% of the trace records include a useful player ID. This is because in the default mode, a WMP sends an anonymous pseudo ID. It only sends a useful player ID if the user voluntarily participates in the quality feedback program.

In order to preserve the scale of the data set and not discard 93% of the trace records unnecessarily, we use the following method to classify the anonymous players. Each trace record contains additional client information, including the player’s host public IP address, WMP version/language, OS/version/language, and the web browser/version. For each record with an anonymous player ID, we hash the concatenation of this additional client information. We then consider all records with the same hash as being generated

¹This calculation will *not* work for Multi-Bit Rate (MBR) videos, where each video file contains several independently encoded streams at different quality level (thus bitrates). Nevertheless, there are not many MBR videos on the MSN Video site (less than 1%), so we simply filter out all the requests for such videos.

from the same player. We refer to each such player as a *hash-identified player*, whereas we refer to each non-anonymized player as an *ID-identified player*. To verify that records from different hashes indeed come from different players, we examine the distribution of the number of media files streamed by an ID-identified player and the number streamed by a hash-identified player. If these distributions are similar, we can conclude that the hash-identified players indeed correspond to unique players. Figure 1 shows that these two distributions align very closely, validating this method for classifying the anonymous players.²

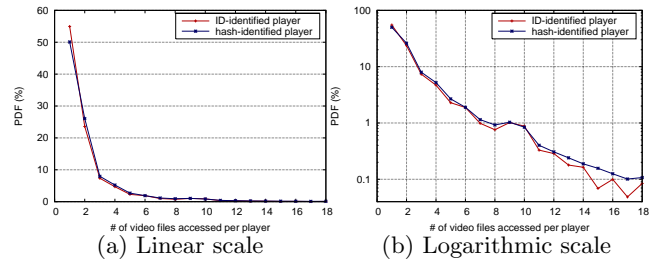


Figure 1: Video file access pattern (April 20)

Having now grouped the trace records by player, we now further group the records by session. As mentioned previously, a single user-video session can generate more than one trace record if the user interacts with the video during playback. We define a *streaming session* as a series of streaming requests from the same player to the same video file, as long as the beginning time of one request is no more than T seconds after the ending time of the previous request. In this paper, we set $T = 10$ s. From the original more than 520 million streaming requests, we obtain over 471 million streaming sessions in total.

2.3 Video Popularity Distribution

The greater the locality of requests to a subset of the videos, the greater the potential benefit for peer-assisted streaming. In this subsection we examine the distribution of video popularity in the trace. To this end, we pick the three days in April 2006 which have the minimum, median, and maximum amount of traffic in the month. The popularity distributions are plotted in Figure 2. First observe that, despite the traffic variation among the three days, the popularity distributions are quite similar. Second observe there is indeed a high-degree of locality. Third observe that the distribution is more skewed than a Zipf distribution. This is likely due to the fact that on any given day, there are several highly popular news and business clips, with each of these clips having roughly the same popularity (explaining why Figure 2 is almost flat for the ten most popular clips).

2.4 User Demand and Upload Resources

In order to further investigate the potential benefits of peer-assisted VoD, we would like to determine the aggregate upload resources of the participating peers, and compare the aggregate upload resources with the aggregate user demand.

²Same users still could map to different hash identifiers, due to DHCP, major software updates, etc. This, however, does *not* affect our analysis in later sections, as we rely on hash consistency during a short session, not across sessions at all.

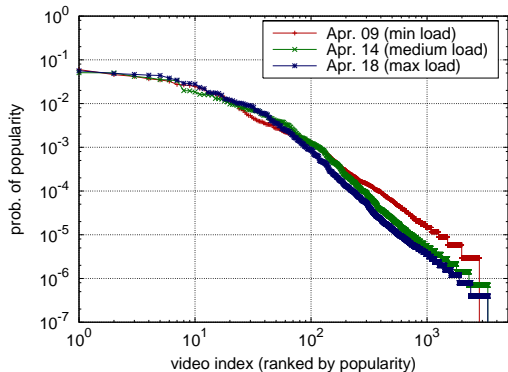


Figure 2: Video popularity distribution

Because the players do not report their upload bandwidths to the media servers, the trace records do not explicitly include any information about client upload capacity. However, whenever a connection is established, the Windows Media Server measures the download bandwidth from the server to the client host. Figure 3 shows the distribution of the measured download bandwidths. Windows Media Server does not distinguish user download bandwidths greater than 3.5 Mbps; thus all measurements greater than 3.5 Mbps collapse to a single point, giving the vertical line shown in Figure 3.

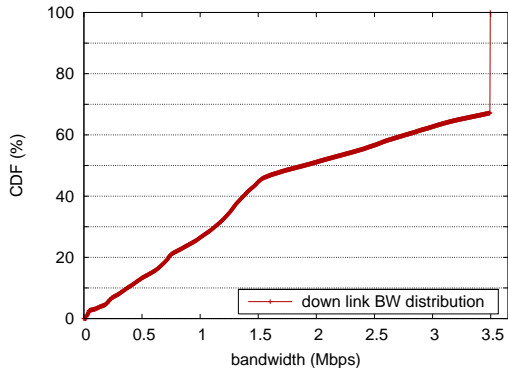


Figure 3: Cumulative distribution function (CDF) of user download bandwidths

Table 1 provides a simple mapping between download and upload bandwidths based on available DSL/Cable offerings. From this table and from the measured upload bandwidths, we infer the upload bandwidth for each user (ID-identified and hash-identified) in the trace. We can also use this technique to estimate the distribution of user types that access MSN Video. As shown in Table 1, over 50% of the MSN Video users have residential broadband connections (DSL or Cable), and 37% of the users have download rates in excess of 3.5 Mbps.

	modem	ISDN	DSL1	DSL2	Cable	Ethernet
download	64	256	768	1500	3000	> 3000
upload	64	256	128	384	768/384	768
share (%)	2.8	4.3	14.3	23.3	18.0	37.3

Table 1: User bandwidth breakdown (kbps)

Note that we are making very conservative assumptions about users' upload bandwidth. For instance, many users with download bandwidths greater than 3.5 Mbps are likely on university and corporate networks and therefore have much higher upload bandwidths than the 768 kbps assumed here. On the other hand, a benign peer-assisted VoD system may not want to fully exploit the available bandwidth of the high capacity users, as doing so might deter their participation.

The healthiness of any peer-assisted delivery system relies critically on users' willingness to contribute their upload bandwidths. Incentive can be built into the delivery mechanism itself, as in BitTorrent, but not necessarily have to. It can be provided through other means. For example, content providers may offer users with options, where they can choose *not* to act as peers and get content directly from a server, but at low quality, or they can opt for peer-assistance and get much higher quality. Providers can also make available some premium content for free or reduce the amount of advertisements users have to watch, to encourage the participation in peer-assistance. Moreover, as will become clear later, the peak server bandwidth consumption matters most to content providers. Hence, they might also offer dynamic incentives (e.g. Microsoft Points, which can be used to purchase music, videos, and games, etc.) to encourage more sharing during peak hours. As long as peer-assistance has limited interference with users' other networking activities, we believe users in general will *not* object to share, given the fact that they are obtaining valuable services, as well as direct values in certain scenarios.

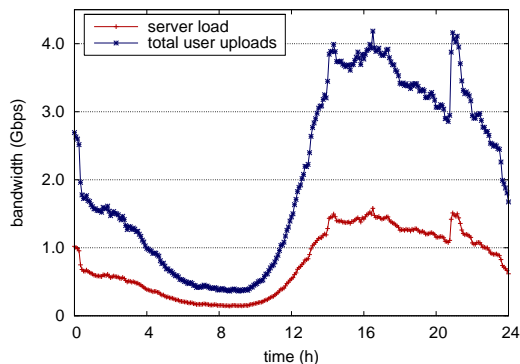


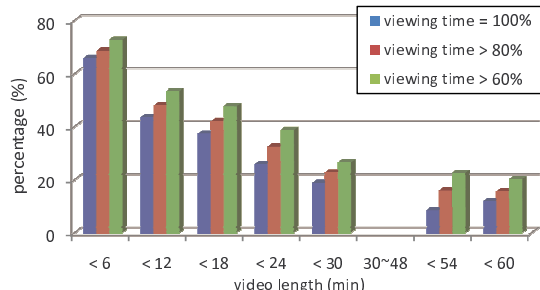
Figure 4: Aggregate user demand and upload resources (April 18)

Having developed a reasonable heuristic for estimating the upload bandwidth of a user, we can determine from the trace records the aggregate upload bandwidth and aggregate demand at the server for any instant of time. Figure 4 presents the evolution of the aggregate demand and the aggregate upload capacity for a 24-hour period on April 18 (the day with maximum traffic in April). As we would expect, the demand greatly fluctuates during the day, with relatively little demand placed on the servers in the early morning. However, the large margin by which the total upload resources exceeds the user demand is beyond expectation, suggesting that a peer-assisted VoD solution might perform very well.

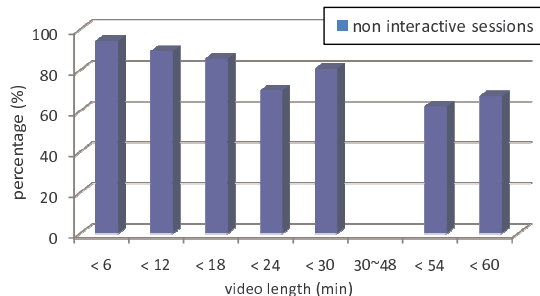
2.5 User Interactivity

When a user selects a video for streaming, the user does

not necessarily continuously watch the video from beginning to end. The user may quit watching it before completion, may pause and then resume, may skip segments of the video, may fastforward the video, and so on. It is important to understand this interactivity while considering peer-assisted solutions for VoD. To this end, we examine user interactivity in the entire month of April 2006.



(a) Percentage of sessions viewing more than 60%, 80% and 100% of the video



(b) Percentage of sessions without gaps

Figure 5: User behavior (April)

Figure 5(a) shows the percentage of streaming sessions that view more than 60%, 80%, or 100% of the video for different video lengths. We first observe that users generally view large fraction of short videos, but less than 20% of the users view more than 60% of the video for videos longer than 30 minutes. We will see in Section 4 that the fraction of sessions that start at the beginning of a video and have no interactivity (but may quit before the end of the video) is an important factor in the potential success of a peer-assisted VoD system. Figure 5(b) shows this fraction for different video lengths. We see that, generally, a large fraction of the users view videos without interactivity. For shorter videos (under 30 minutes), about 80% of the sessions do not have interactivity; for the longer videos, about 60% do not have interactivity.³

2.6 Service Evolution

VoD over the Internet is in its infancy and will certainly evolve over the upcoming years. In designing a new architecture for VoD applications, we should bear in mind major application trends. The year 2006 was a year of intense activity in the Internet VoD space, with major players such as YouTube, Google Video and MSN serving tens of thousands

³There are no videos of length 30~48 mins. Also note that a large number of videos are not very longer, typically between 5 to 15 minutes. When full length episodes grow more popular, user interactivities are likely to increase [3].

of years of content. Over the upcoming years, we not only expect aggregate viewing hours to increase, but also video quality to increase (by increasing the bit rate of the videos). We now explore the growth in demand and video bit rate that occurred during the nine-month period of this study.

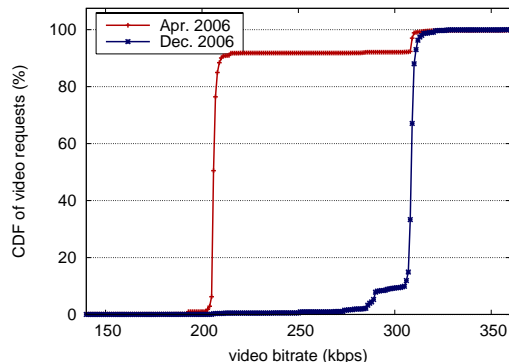


Figure 6: Quality evolution

Figure 6 plots the distribution of the bitrates for the MSN requested videos for two months: April 2006 and December 2006. In April 2006, the majority of the requests were for videos encoded at 200 kbps. Nine months later, the majority of the requests are for videos encoded at 320 kbps. This implies that MSN Video increased the bit rate of its videos by over 50% during the nine-month period. In the upcoming years, it is likely that video bit rates will continue to increase at a rate faster than average peer upload bandwidths. This would have the effect of bringing the two curves in Figure 4 closer together. We will see in Section 3 that when these two curves are close, particular care must be taken in designing a peer-assisted VoD system.

	April	December	up (%)
total requests	41.1 M	64.7 M	57.4
unique users	9.03 M	12.02 M	33.1

Table 2: Traffic evolution

Perhaps partially due to the service quality upgrade, together with the general trend that more users are spending more time watching Internet videos, the amount of traffic served by MSN Video servers increased substantially over the period, as shown in Table 2.

2.7 95 Percentile Rule

Typically today an ISP (or a CDN) bills a customer (such as a VoD provider) for bandwidth usage using the so called *95 percentile rule*. In this paper we will use this rule as the basic metric for measuring the cost to the service provider. The 95 percentile rule is in essence a rule that charges the service provider each month according to the service provider's peak bandwidth usage, but neglecting in the calculation "anomalous periods" during which bandwidth usage is unusually high. Specifically, the 95 percentile rule works as follows. The average server bandwidth is measured every 5 minutes within each month. These bandwidth measurements over a month form a set of values, and the 95 percentile value is the smallest number that is greater than 95% of the values in the set.

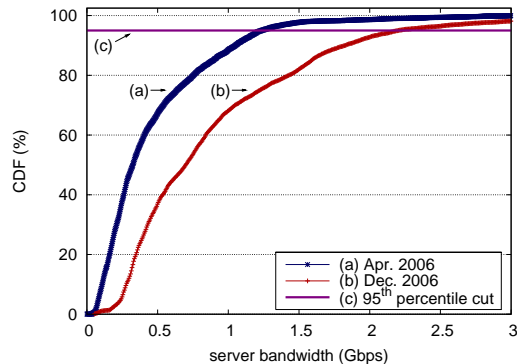


Figure 7: 95 percentile rule: CDF of 5-minute averages in April and December

Figure 7 shows the 95 percentile value for two different months: April 2006 and December 2006. Using this rule, we see that the bandwidth cost increases from 1.23 Gbps in April to 2.20 Gbps in December. Hence, because of the video quality upgrade and the increased popularity of the service, the operational cost is increased by 78.5%. Looking forward, when more users spend even more time watching Internet videos, they are going to demand longer videos and better quality. Providing such services to meet the demand with a client-server architecture is going to be very costly (if not prohibitive).

3. THEORY OF PEER-ASSISTED VOD

In this section we develop simple mathematical models for peer-assisted VoD. The purpose of these models is to gain broad insights into the design of peer-assisted VoD. In the subsequent sections, we will combine the modeling insights with trace-based simulations, enabling us to assess the full potential of P2P solutions.

We focus on a single on-demand video of rate r bps. Users watching the video will assist in the re-distribution of the video to other users. We classify users according to their upload link bandwidths. In particular, let M denote the number of user types, where a type m user has upload link bandwidth w_m . Users arrive at the system in a Poisson process with parameter λ . Denote by p_m the probability that an arrival is a type m user. Using the property of the compound Poisson process, the above user arrival model is the same as if the m user types arrive as independent Poisson processes with parameters $p_m\lambda$, $1 \leq m \leq M$. Naturally, the average upload bandwidth of an arriving user is $\mu = \sum p_m w_m$. Because a user may quit the video before it ends, skip portions of the video, pause the video, or perform other interactive actions, the time during which a user remains in the system is random. Denote a user's expected sojourn time in the system as σ .⁴

It follows from Little's law that in steady state the expected number of type m users in the system is given by $\rho_m = p_m \lambda \sigma$. Thus, in steady state, the average demand is $D = r \sum \rho_m = r \lambda \sigma$ and the average supply $S = \sum w_m \rho_m = \mu \lambda \sigma$. We say that the system is in the *surplus mode* if $S > D$; in the *deficit mode* if $S < D$; and in the *balanced mode* if

⁴The model is independent of actual collected traces and thus allows us obtain broad insights.

$S \approx D$. Note that the system is in the surplus mode if $\mu > r$, and in the deficit mode if $\mu < r$. It is important to note that even if a system is in surplus mode, at any given instant of time, the server may need to be active and supply video to peers. This is because (i) although on average the system is in the surplus mode, due to peer churn, at a given instant of time the supply may become less than the demand; and (ii) it may not be possible to use all of the supply bandwidth at any given instant of time. The latter point will become more clear in the next subsection.⁵

3.1 No-Prefetching Policy

Using this model, we want to explore the potential for peer-assisted VoD. As we will see, a peer can redistribute video to other peers in many different ways. We first consider the simplest scheme, which we call the *no-prefetching policy*. Under this policy, each user downloads content at the playback rate (that is, at rate r) and does *not* prefetch content for future needs. At any given instant of time, the user may be downloading from multiple peers as well as from the server. In order to see the forest through the trees and understand the essential features of peer-assisted VoD, for the remainder of this section we assume that each user views the video without gaps. (As shown in Figure 5, this assumption is consistent with the behavior of most users.) In particular, in this section we assume that each user watches the video continuously from the beginning of the video until some stopping point, and then leaves the system. The stopping point may be the end of the video or earlier.

Suppose at a particularly instant of time there are n users in the system. Order these n users so that user n is the most recent to arrive, user $n - 1$ is the next most recent, and so on. Thus user 1 has been in the system the longest. Let u_j , $j = 1, \dots, n$, be the upload bandwidth of the j^{th} user and its probability be $p(u_j)$. Recall that user j is of type m with probability p_m , so $p(u_j = w_m) = p_m$. Let the state of the system be (u_1, u_2, \dots, u_n) and the rate required from the server be $s(u_1, u_2, \dots, u_n)$. Since there is no prefetching, the demand of user 1 can only be satisfied by the server, which is the video rate r . Then, the demand of user 2 will be satisfied first by user 1 and then the server if u_1 is not sufficient. The demand of user 3 is satisfied first by user 1, user 2 and then the server, and so on. Thus

- For $n = 1$, we have $s(u_1) = r$.
- For $n = 2$, we have $s(u_1, u_2) = r + \max(0, r - u_1)$.

In general, for a given state, the rate required from the server is

$$s(u_1, u_2, \dots, u_n) = \max_{1 \leq j \leq n} (r + \max(0, (j-1)r - \sum_{i=1}^{j-1} u_i)). \quad (1)$$

For this no-prefetch policy, it can be shown that the average additional server rate needed is given by

$$s = \sum_n \frac{(\lambda\sigma)^n}{n!} e^{-\lambda\sigma} \sum_{u_j} p(u_1, u_2, \dots, u_n) s(u_1, u_2, \dots, u_n), \quad (2)$$

⁵The concept of operating modes here is similar to the definition of resource index in [15]. But different from live streaming, VoD is unique in prefetching, which we elaborate later in the section.

where

$$p(u_1, u_2, \dots, u_n) = \prod_{1 \leq j \leq n} p(u_j).$$

Although this result is not in closed form, the result can be used to rapidly and accurately determine s using Monte Carlo summation. In the Monte Carlo summation technique, we don't need to simulate the user arrivals and departures and wait for the system to enter steady state, as typically the case in discrete-event simulation. Instead, we simply generate random n and u_j 's, and repeatedly evaluate s using the expression (2).

Detailed analysis is reported in our early work [4]. Here, we briefly summarize the key observations. First, when the supply S is greater than the demand D by a substantial margin (i.e., there is sufficient average surplus in the system), the server rate is very close to the video bit rate r and does *not* increase as the system scales (i.e., as the number of users grows). Second, even with little average surplus, the simple no-prefetching approach can greatly reduce the server rate. Third, when operating in the surplus mode, the service provider can increase the quality of the video (that is, increase r) without incurring significant increases in average server bandwidth *even as the system scales to a large number of concurrent users*. At bit rates below 1.0 Mbps, each bit rate increment brings noticeable difference in perceived quality. Laws of competitive economics dictate that the service provider should therefore increase service quality when in the surplus mode, thereby driving the operation of the system into the balanced mode. Forth, when the supply S is less than the demand D by a substantial margin, the server rate almost equals to $D - S$. This means when the system is in this high-deficit mode, users do *not* need to adopt sophisticated prefetching approaches. Fifth, as the system moves from the balanced mode to the deficit mode, the server resources increases dramatically, particularly when there is a large number of users. Finally, the no-prefetching policy does *not* perform well in the balanced mode, where we expect most Internet VoD systems to operate in the future due to basic economic motivations. This unsatisfactory performance will be particularly pronounced when the average number of users watching the video is small. This motivates us to consider more sophisticated prefetching policies.

3.2 Bandwidth Allocation Policies for Prefetching

The non-prefetching policy described in the previous subsection never uses more upload capacity than necessary to meet the instantaneous demand of the peers. But when the instantaneous aggregate upload capacity exceeds the instantaneous aggregate demand, there is typically surplus upload capacity beyond what is needed to satisfy the demand. This surplus upload capacity can be used to distribute to the peers future content, creating a reservoir of prefetched content that can be exploited when the system shifts into a deficit state. Such prefetching schemes are particularly attractive when the system operates in the balanced mode, where the system fluctuates between surplus and deficit states.

The primary goal of peer-assisted VoD is to save bandwidth costs at the server. Thus, in our prefetch schemes, the sever never sends prefetched content, and is only used to fulfill the current demand. Also, when a user has a reservoir of prefetched content, it can drain its reservoir before it

requests new data. Hence, the demand of each user varies depending on its buffer level, as opposed to the constant demand with the no-prefetching policy. Also observe, that if a peer succeeds to prefetch to the end of the video, then this peer's demand becomes zero and stops receiving prefetched content.

When allowing for prefetching, an important question arises in how to allocate the instantaneous surplus upload capacity among the peers in the system. Should we devote all the surplus capacity to one peer, rapidly building a reservoir for that peer while neglecting the other peers? Or should we try to equally allocate the surplus bandwidth, building small reservoirs of content at each of the peers? There are actually infinitely many allocation schemes. In this section, we consider two schemes which, intuitively, should perform well. In fact, at the end of this section we show that the two schemes are nearly optimal.

Our first allocation scheme is the *water-leveling policy*, which aims to equalize the reservoir levels of prefetched content across all the peers. Thus, when there is aggregate surplus capacity, if one peer has less prefetched content than the others (because, for example, it has just arrived at the system), then to the extent possible, the aggregate surplus capacity is channeled to this peer. Once the reservoir level of this peer reaches that of the other peers, then an attempt is made to equally distribute the surplus capacity among all the peers, thereby maintaining uniform reservoir levels across all peers.

Alternatively, we have also considered a *greedy policy*, where each user simply dedicates its remaining upload bandwidth to the next user right after itself. The greedy policy works in the following two steps: First, all users are scanned through to determine the server rate that is needed to satisfy the real-time demands. During this process, the remaining bandwidth at each user is recorded. Second, each user is scanned through again and allocates as much bandwidth as possible to the subsequent user. Please refer to our early work [4] for the details of the two policies.

3.3 Lower Bound

We have just described two allocation policies for prefetching: water-leveling and greedy. Ideally, we would like to identify the optimal policy, that is, the dynamic allocation policy that minimizes the average server bandwidth (or better yet, the 95% level at the server). Alas, this is a difficult scheduling problem. Instead, we determine a lower bound for the average server bandwidth and investigate how close water-leveling and greedy are to the lower bound.

We first point out an obvious lower bound. Recall that S is the average supply and D is the average demand. If $D > S$, then on average the server must at least make up the difference, that is, the average server rate is at least $\max(D - S, 0)$. This trivial bound turns out to be too loose to be useful in most circumstances.

We now outline how a more insightful lower bound can be obtained. The basic idea is to remove the ordering restrictions from the allocation policies. That is, we suppose that a peer can feed content to any peer, not just to the peers that arrived after it. In this modified system, when there are n peers in the system with upload capacities u_1, u_2, \dots, u_n , then the entire aggregate capacity $u_1 + u_2 + \dots + u_n$ can be used, irrespective of arrival ordering of the peers. It is clear that water-leveling is optimal for this modified sys-

tem. Thus, this modified system, which always makes full use of the aggregate upload capacity and which allocates bandwidth according to the water-leveling policy, will have lower average server bandwidth than *any* prefetch policy for the original system. If a prefetching policy is close to this lower bound, then it is nearly optimal.

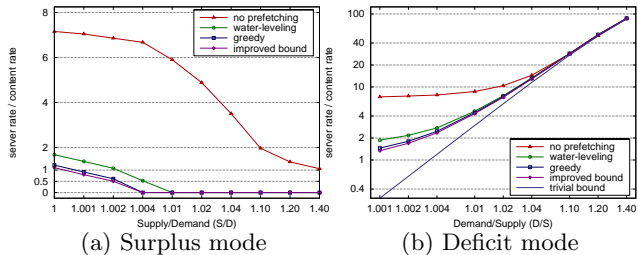


Figure 8: Comparison of three prefetching policies

3.4 Simulation Results

We developed a discrete-event simulator to study the performance of the water-leveling and greedy policies and to compare those policies to the no-prefetching policy and the lower bound. We focus on the balanced mode, as that is where there is the greatest potential for improvement with the prefetch policies. It is also where we expect VoD systems to operate in the future.

We fix the user arrival rate at $\lambda = 1$ and vary the supply-demand ratio between 1/1.4 to 1.4. Figure 8 shows the simulation results. We make the following key observations:

- When the system operates in the balanced mode, prefetching can provide significant improvements, reducing server bandwidth requirements by a factor of three or more. Note that the domain of the balanced mode becomes wide for systems with a relatively small number of users.
- The greedy policy does slightly better than the water-leveling policy.
- Both the greedy and water-leveling policies are very close to the lower bound. Thus, both of these policies are nearly optimal.

In the subsequent section, we will use the results of this model to guide the trace simulation study.

4. REAL-WORLD CASE STUDY

The previous section presented a theoretical model that exposed many of the fundamental issues of peer-assisted VoD. In this section, we use the MSN Video trace data described in Section 2 to gain critical complementary insights. Although the traces were generated from a client-server deployment, we use them here to drive simulations of peer-assisted VoD.

4.1 Trace Analysis for the Two Most Popular Videos

We are particularly interested in the impact of early departures and peer interactivity (skipping video segments) on the performance of peer-assisted VoD and prefetching policies. We consider three cases in our trace analysis: (i) all

users watch the entire video without departing early and without interactivity (that is, re-positioning in the content); (ii) preserving early departures but still ignoring interactivity; and (iii) preserving the original trace with both early departures and user interactivity.

We first consider case (i), that is, all users watch the entire video. To this end, we modify the trace data so that each arriving user stays in the system for the length of the video, and disregard all skipping request records that do not start from the beginning of the video. We consider the two most popular videos in April 2006. Figure 9 shows the server resources used by the pure client-server deployment is shown as the *no P2P* curve. Because the server rate is proportional to the number of users watching the video, the no-P2P curve also reflects the user demand for the video over time.

The request patterns on these two videos are strikingly different. The most popular video, which we call the *gold stream*, was released on April 5 and quickly attracted a large number of requests. However, the demand for this video declined very quickly, with only occasional views 5 days after it was released. The second most popular video, the *silver stream*, was popular throughout the entire month. We observe that the demand for this video went through a daily peak and valley cycle. Interestingly, the demand also has a 7-day cycle as well, where the valley matches nicely with weekend times.

We use these traces to drive the simulation and study the performance of the proposed policies. Because our model in Section 3 indicated that the greedy policy performs better than the water-leveling policy, and is generally close to the lower bound, we focus here on the greedy policy for the peer-assisted deployments. In Figure 9, we plot the server rate with peer assistance for two cases: at the current quality level and at $3\times$ the current quality level. The $3\times$ video rate pushes the system from the surplus mode into the deficit mode.

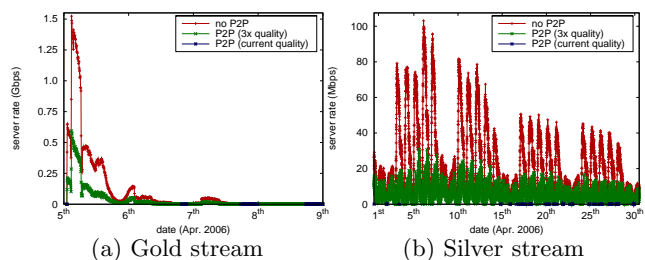


Figure 9: Dynamics of required server resources (all users watch entire video)

We make the following observations: First, if a peer-assisted distribution system had been used instead of the client-server system, the server rate would have been dramatically reduced. In fact, Figure 9 shows a potential 1000-fold server rate reduction! Second, for the P2P deployment at the current quality level, typically no server resources are needed. The occasional traffic at the server occurs when the demand for the video enters valleys, which correspond to small numbers of concurrent users in the system. This conforms with our analysis. When the number of concurrent users is small, there is greater (normalized) variance in the upload capacity, so that peer-assisted VoD is more likely to

run into a temporary deficit states that require server participation. Third, for these two videos, MSN Video is clearly operating in the surplus mode due to the relatively low bitrates of the videos. When a peer-assisted VoD solution is deployed, we can easily offer much higher streaming quality (e.g. tripling the bitrate) and still trim the server rate significantly. Finally, peer-assistance can be beneficial for both flash crowd (gold stream) and long-lasting (silver stream) videos.

serv. (Mbps)	gold stream			silver stream		
	N.P.	greedy	bound	N.P.	greedy	bound
client-server	41.1			66.8		
cur. qual.	.23	.18	.17	.26	0	0
3x qual.	12.5	11.9	11.9	14.5	14.2	14.1

Table 3: 95 percentile rule for client-server and peer-assistance (N.P. no-prefetching).

Table 3 presents the performance of peer-assisted VoD in the context of the 95 percentile rule. Results are given for the current quality scaling (surplus mode) and the 3x bit-rate scaling (deficit mode). We observe dramatic improvements going from client-server to peer-assistance with no-prefetching, and then further improvements going from no-prefetching to the greedy prefetching policy in the deficit mode. The performance of the greedy policy is also very close to the bound. Note that the server bandwidths of the gold stream are actually less than those of the silver stream, because there was little demand 5 days after the release of the gold stream.

4.2 Impact of Early Departures

In this subsection, we again use the April traces but now preserve early departures when they occur. Hence, the duration of each session now varies, as opposed to all sessions having the same video length as in the previous subsection. So as not to introduce too many factors at once, we still disregard user interactivity in this subsection.

We also focus on the balanced mode in this subsection. By scaling the video bitrate, we can easily drive the system from the surplus mode, through the balanced mode, to the deficit mode. Using the silver stream as an example, Table 4 shows server rates, averaged over the entire month of April, for both the no-prefetching and the greedy prefetching policies. From this table we observe that – even with early departures – peer-assistance can provide a dramatic improvement in performance. We also observe that prefetching continues to provide improvements over non-prefetching, particularly in the balanced mode (scaling 1.8 to 2.6).

Bitrate scaling	1.0	1.4	1.8	2.2	2.6	3.0
Client-server	36.0	50.4	64.8	79.2	93.7	108.1
No-prefetching	.26	.40	.60	1.01	2.55	7.60
Greedy	0	.01	.18	.54	2.09	7.19
Lower bound	0	0	0	.22	1.81	7.13

Table 4: Server rates (in Mbps) under different system modes with early departures

4.3 Impact of User Interactivity

As described in Section 2, user interactivity is not uncommon in Internet VoD, particularly for long videos. For example, we saw that for videos over 30 minutes in length, roughly 40% of sessions contained some interactivity. This subsection, we assess the impact of interactivity on peer-assisted VoD.

When there is user interactivity, a user might have holes in its buffer; thus a user may not be capable of sharing all content up to its current playback point. It is possible to keep track of all buffer segments at all users, and simulate the system based on the exact buffer status; but such simulation would be computationally demanding and may even become prohibitive, given the size of our data set. Instead, we choose to examine two extreme approaches, which intuitively should serve as good bounds for the actual performance. The first approach, which we refer to as the *conservative approach*, sets the user upload bandwidth to zero after interactivity. As such, since a modified user no longer serves content after interactivity, the holes in its buffer will *not* matter at all. The second approach, which we refer to as the *optimistic approach*, simply assumes there is no hole in the user’s buffer even after interactivity and therefore the user can upload any content being requested. Clearly, the actual performance will lie between these two bounds.

Using the traffic on April 18 as an example, we plot the total available user upload bandwidth corresponding to both approaches in Figure 10. (Note that two out of the three curves here are the same as those in Figure 4.) As the

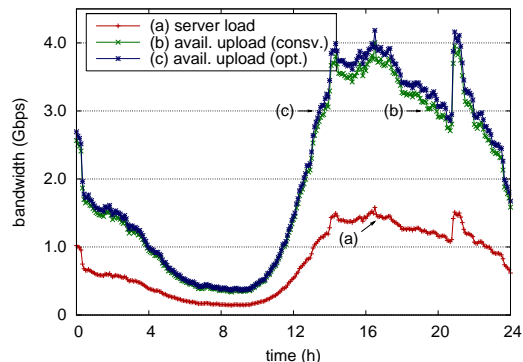


Figure 10: User demand and upload resources (April 18)

total available bandwidth gap between the two approaches is not significant, intuitively, the loss of upload bandwidth due to interactivity is not significant. Thus, the results of the previous subsection (accounting for early departures) are also representative for interactivity.

4.4 All Things Considered

Having explored the impact of early departures and user interactivity, we now investigate the total server bandwidth cost for the month of April 2006. During this month, MSN made available more than 12,000 videos. The multiplexing effect across the 12,000+ videos is of interest, as one video’s peak might happen to be another video’s valley. Here we use the 95 percentile criterion to measure the server bandwidth cost. We once again examine three deployment scenarios: client server; using P2P without increasing quality;

and using P2P with 3 times quality. For each of the P2P deployments, we use the greedy prefetching policy.

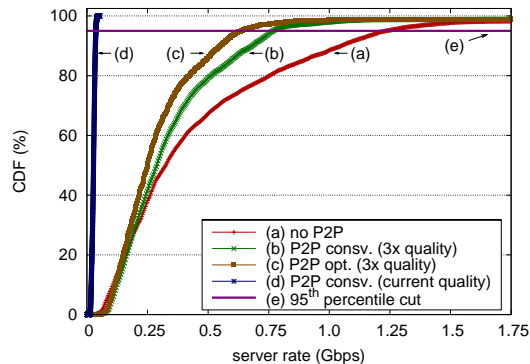


Figure 11: CDF of 5-minute average bandwidths for the month of April using all videos

Figure 11 shows the cumulative distribution of the five-minute average bandwidths for the month of April using all 12,000+ videos. The results from both the conservative and optimistic approaches are presented. We omit the curve for the optimistic peer-assisted approach at the current quality level, as it aligns very closely with the curve of the conservative approach. The savings using the 95 percentile rule are summarized in Table 5.

	consv. approach		opt. approach	
	serv. rate	saving (%)	serv. rate	saving (%)
Client-server (Gbps)	1.23			
cur. qual. (Mbps)	37	97.0	33	97.3
3x qual. (Mbps)	770	37.6	636	48.5

Table 5: 95 percentiles for the month of April 2006

By using peer assistance, even under very conservative assumptions (the conservative approach), the server bandwidth can potentially be reduced by 97.0% at current quality level. Alternatively, peer-assisted solution can triple the video rate (with a corresponding improvement in quality) and still trim the server bandwidth by 37.6%.

It is also interesting to examine the contribution of individual videos in the final server bandwidth costs. To this end, we rank the 12,000+ videos by their popularity and then classify them into 4 groups: 1) *group A* contains video 1-10; 2) *group B* video 11-100; 3) *group C* video 101-1000; and 4) *group D* the rest 11,000+ videos. We provide the 95 percentile value for each group in Table 6. We make the

serv. rate (Mbps)	# of videos	no P2P	cur. qual.	3x qual.
group A (1-10)	10	295	0.7	148
group B (11-100)	90	523	3.3	296
group C (101-1000)	900	523	16.3	331
group D (1000+)	11396	76	21.3	112

Table 6: Server bandwidth cost breakdown across videos

following observations:

- The top 10 videos contribute significantly to the fraction of server bandwidth cost under the pure client-server model, but much less so with peer-assistance.

This essentially echoes the well-known scalability of peer-to-peer applications: the more popular the content is, the easier it is to distribute with peer assistance.

- The top 100 videos account for more than half of the server bandwidth cost for client-server distribution. There are again significant gains when using peers to offset the server load. Nevertheless, it is not sufficient to only deal with these top 100 videos, which leads to the next point.
- The videos in group C also use a large portion of the server bandwidth, suggesting that a large number of videos offered by MSN Video are very popular.
- The gain of peer-assisted VoD diminishes for the unpopular videos. For example, when we triple the quality of all videos in group D, the proportional server bandwidth increment is still quite significant even with peer-assistance. Thus we can conclude that it is much simpler (in terms of system design and maintenance) to rely on the pure client-server model to delivery such videos.

Note that the sum of any two values in the table does *not* have an accurate meaning, due to the 95 percentile rule. For instance, the 95 percentile value for the top 100 videos (not using P2P) is 709 Mbps, which is much smaller than the sum of the corresponding values from group A and group B. . Nevertheless, this breakdown into 4 groups offers reasonable relative comparison and important insights.

4.5 Service Cost Evolution

Finally, we examine the impact of service evolution on peer-assisted VoD. As described in Section 2, from April to December, the server bandwidth cost increased by 78.5% for the client-server deployment due to the video quality upgrade and the increased popularity of the service. Table 7 shows that if a peer-assisted VoD solution had been instead deployed between April and December, then the server bandwidth bill of December could have potentially been trimmed by 93.6%, compared to that of April.

	client-server VoD	peer-assisted VoD	savings (%)
April	1.23 Gbps	36.9 Mbps	97.0
December	2.2 Gbps	79.4 Mbps	96.4

Table 7: Server costs for April and December

5. THE IMPACT OF PEER-ASSISTANCE ON INTERNET SERVICE PROVIDERS

Up until this point, we have described peer-assisted VoD purely from the perspective of content providers. The main objective was to maximally utilize participating peers' upload resources so as to reduce the server bandwidth costs. However, when peers are sending large amounts of data to each other, much of this data may go from one Internet Service Provider (ISP) to another and in turn incur non-negligible costs for the ISPs. In this section, we take the concern of ISPs into consideration and explore how peer-assisted VoD can strike a balance between the conflicting interests of reducing the VoD provider's server cost and reducing the P2P cross traffic among ISPs.

5.1 Economics of ISP Relationships

The Internet consists of thousands of ISPs, which operate at very different scales and serve very different roles. Some ISPs provide Internet access to end-users and businesses, while others provide access to ISPs themselves. The relationships between ISPs can be summarized into 3 categories [5]: 1) *transit relationship*, which refers to one ISP purchasing Internet access from another ISP and paying for the bandwidth usage. This is also called customer-provider relationship; 2) *sibling relationship*, which refers to the interconnection among several ISPs belonging to the same organization. Even though each ISP might be managed separately from the perspective of network administration, traffic exchange among them does *not* involve any payment; and 3) *peering relationship*, which refers to ISPs pairing with each other. Peering ISPs can exchange traffic directly, which would otherwise have to go through their providers. This is a common relationship adopted to lower ISPs' payments to their own providers. To a certain extent, the traffic exchanged between two peering ISPs is free. However, when the traffic becomes highly asymmetric, one party *will* start charging the other based on bandwidth usage [6].

Based on the ISP relationships, ISPs can be grouped together to form economic entities, whereby no payment is involved for traffic within an entity but traffic crossing entity boundaries *does* incur payment. Based on the sibling and peering relationships, such economic entities can be formed at two levels: 1) *sibling entity* includes all ISPs that are siblings to each other; and 2) *peering entity* includes not only all siblings, but also all ISPs that are peering with each other. Note that a peering relationship is in general not *transitive*; however, in peer-assisted VoD, it's reasonable to assume that data can flow between two ISPs without direct peering relationship, as long as there exists a third party ISP which peers with both of these two ISPs. This is because all peers in the third party ISP can essentially help to tunnel the traffic.

5.2 Without ISP-Friendly Peer-Assisted VoD

When a peer-assisted VoD solution is deployed without considering the economics of ISPs, intuitively, there will be a significant amount of traffic crossing entity boundaries. We now use the MSN Video trace to evaluate the potential costs of peer-assisted VoD to ISPs. Each trace record contains the public IP addresses of peers, which can be mapped to ISPs using tools like the ASFinder in the CoralReef suite [7]. Then, based on an inferred AS relationships dataset [8], we can group ISPs into economic entities. Again, there are two levels of entities: sibling entities and peering entities. To keep things simple, we adopt the following approach to estimate the traffic crossing entity boundaries. For each user, we categorize into the classes the peers from which the user receives content: 1) from peers within the same entity; and 2) from peers from other entities. We assume, at any time, the ratio of traffic from these two classes is equal to the ratio of upload bandwidths for these two classes. Although not an exact measurement of the crossing entity traffic, we believe this yields a good estimate when peers are randomly paired together, which is a common practice in many deployed peer-to-peer systems.

Figure 12, shows the percentages of traffic contained within entities, as well as those crossing entity boundaries. Clearly, we can see that the majority of the P2P traffic is actually

crossing entity boundaries. This implies that a peer-assisted VoD solution might greatly increase the ISPs' costs, if deployed without any regard for the ISPs. Importantly, we observe that the amount of cross traffic is reduced significantly when the sizes of entities increase. Note that the cross traffic and contained traffic do *not* add up to 100%, as a tiny portion is coming from the server.

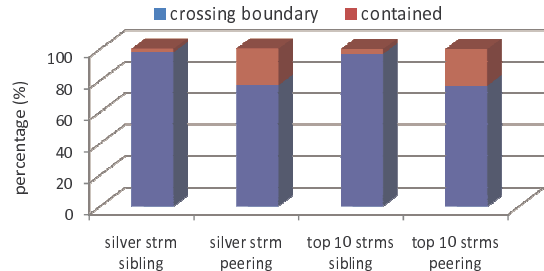


Figure 12: Cross traffic without ISP-friendly peer-assisted VoD

5.3 Pure ISP-Friendly Peer-Assisted VoD

We have just examined the impact of peer-assisted VoD on ISPs, when the solution is solely focused on minimizing the bandwidth costs of the content providers. Now, we consider the other extreme, where we restrict the P2P traffic to be contained within entity boundaries. We refer to this approach as *ISP-friendly peer-assisted VoD*. In this scenario, due to the rigid entity partitioning, the distribution of one video becomes several separate distributions, one for each entity watching the video. Using the silver stream as an example, with either sibling or peering entity partitioning, we observe more than 5,000 distinct distributions.

Intuitively, when an entity contains few peers, the sharing becomes more difficult as well, and the server bandwidth is increased accordingly. Table 8 shows the server bandwidth costs when peers never share across entity boundaries. Compared to the scenario where the ISP issue is completely ignored (Figure 12), the savings to the ISPs are significant. Nevertheless, compared to the client-server model, ISP-friendly peer-assistance still provides more than 50% savings, which is surprisingly good. This result also suggests that the inherent ISP clustering among the users justifies efforts for further exploration.

	no P2P	sibling partition	peering partition
silver	39.0	19.6	15.8
top 10	295.2	90.3	75.1

Table 8: Server bandwidth (in Mbps) in an ISP-optimized scenario

For practical peer-assisted VoD, it is important to strike a balance between these two extremes. In doing so, deployments will hopefully provide significant reductions in bandwidth costs to content providers without generating unacceptable levels of traffic across ISP boundaries.

Finally, we point out that the inferred AS relationships from CAIDA [10], which we used in this study, may be very conservative. For instance, we examined the following two ISPs with AS numbers: AS3598 and AS8075. The inferred

relationship between them is classified as the customer-provider relationship. Given that, peers from these ISPs will be separated into two different entities during our above evaluation. However, these two ISPs both belong to Microsoft Corporation, and it is likely no payment is involved for the traffic exchanged between them. Historically, the inference of AS relationships has been conservative [9], which is largely due to the fact that ISPs are unwilling to share their sibling and peering relationships. With peer-assisted VoD looming on the horizon, hopefully ISPs will be motivated to share their sibling and peering information so that truly ISP-friendly peer-assisted VoD solutions can eventually prevail.

6. RELATED WORK

Peer-assisted content delivery has attracted many research efforts, covering several application areas: file distribution, live streaming and on-demand streaming.

The popular peer-assisted protocol BitTorrent [11] has been the focus of several recent studies that address the performance of its underlining protocol. Qiu and Srikant [14] developed a fluid model to obtain analytical insights. Bharambe et al. [13] evaluated BitTorrent performance through extensive simulations. Legout et al. [12] have shown BitTorrent to be close to optimal under a variety of conditions. Whereas peer-assisted file distribution is now fairly understood, peer-assisted on-demand streaming is still an emerging area.

In contrast with peer-assisted file distribution and peer-assisted live streaming, both of which have already enjoyed large-scale deployments, peer-assisted VoD is still in its infancy. To our knowledge, there is no documented deployment of a large-scale VoD system to date. There has, however, been a number of proposals for peer-assisted VoD. Cui et al. [20] proposed oStream, which extends application-layer multicast to include buffers in the peer nodes to support VoD. Hamra et al. [19] proposed a tree based approach, where new incoming peers are connected to the nodes in the tree based on their bandwidth availability. Annapureddy et al. [22] studied using network coding in the VoD scenario. Li proposed PeerStreaming [21] to utilize erasure resilient coding (ERC) and reduce the overall cache requirement. Guo et al. [18] proposed to segment videos into small sub-clips and then use a BitTorrent-like approach to share these sub-clips. Additionally, Xu et al. [16] and Tu et al. [17] considered queuing performance models for media streaming systems. None of these work, however, investigates the performance of peer-assisted VoD using real-world traces. Moreover, none of these work considers the important issues of prefetching, ISP-friendly considerations, and balanced mode operation.

7. CONCLUSION

We consider the design and potential benefits of peer-assisted video-on-demand. Using the nine-month MSN Video trace, we report on key observations of the characteristics from such a large scale VoD service. A theory is presented to explore many of the fundamental issues of peer-assisted VoD. With peer-assistance and prefetching, we show the enormous potential cost savings to content providers. We also examine the costs that peer-assisted VoD might place on local ISPs and explore how these costs can be minimized.

8. REFERENCES

- [1] L. Gomes, "Will All of Us Get Our 15 Minutes On a YouTube Video?" *Wall Street Journal*, Aug. 30, 2006.
- [2] "Windows Media Services SDK Client Combination Logs," <http://msdn2.microsoft.com/en-us/library/ms741431.aspx>.
- [3] B. Cheng, X. Liu, Z. Zhang, and H. Jin, "A Measurement Study of a Peer-to-Peer Video-on-Demand System," *IPTPS*, Bellevue, WA, Feb. 2007.
- [4] C. Huang, J. Li, and K. W. Ross, "Peer-Assisted VoD: Making Internet Video Distribution Cheap," *IPTPS*, Bellevue, WA, Feb. 2007.
- [5] W. B. Norton, "The Evolution of the U.S. Internet Peering System," *white paper*, Equinix Inc., 2003.
- [6] W. B. Norton, "Internet Service Providers and Peering," *white paper*, Equinix Inc., 2001.
- [7] CAIDA. CoralReef suite. <http://www.caida.org/tools/measurement/coralreef>.
- [8] CAIDA. AS Relationships Data. Research Project. <http://www.caida.org/data/active/as-relationships>.
- [9] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Trans. on Networking*, 9(6), Dec. 2001.
- [10] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley, "AS Relationships: Inference and Validation," *ACM SIGCOMM CCR*, 37(1), 2007.
- [11] B. Cohen, "Incentives to Build Robustness in BitTorrent," *Workshop on Economics of P2P Systems*, 2003.
- [12] A. Legout, G. Urvoy-Keller, and P. Michiardi. "Rarest First and Choke Algorithms Are Enough," *ACM IMC*, Rio de Janeiro, Brazil, Oct. 2006.
- [13] A. Bharambe, C. Herley, and V. Padmanabhan, "Analyzing and Improving a BitTorrent Network's Performance Mechanisms," *IEEE INFOCOM*, Barcelona, Spain, 2006.
- [14] D. Qiu, and R. Srikant, "Modeling and Performance Analysis of BitTorrent-like Peer-to-Peer Networks," *ACM SIGCOMM*, Portland, Sep. 2004.
- [15] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points," *ACM SIGCOMM*, Portland, Sep. 2004.
- [16] D. Xu, S. Kulkarni, C. Rosenberg, H.-K. Chai, "Analysis of a CDN-P2P Hybrid Architecture for Cost-Effective Streaming Distribution," *ACM/Springer Multimedia Systems Journal*, 11(4), 2006.
- [17] Y. C. Tu, J. Sun, M. Hefeeda, and S. Prabhakar, "An Analytical Study of Peer-to-Peer Media Streaming Systems," *ACM TOMCCAP*, 1(4), 2005.
- [18] Y. Guo, S. Mathur, K. Ramaswamy, S. Yuy, and B. Patel, "PONDER: Providing Commercial-Quality Video-on-Demand Service Using Peer-to-Peer Network," submitted.
- [19] A. Al Hamra, E. W. Biersack, and G. Urvoy-Keller, "A Pull-based Approach for a VoD Service in P2P Networks," *IEEE HSNMC*, Toulouse, France, Jul. 2004.
- [20] Y. Cui, B. Li, and K. Nahrstedt, "oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks," *IEEE JSAC*, 22(1), 2004.
- [21] J. Li, Y. Cui, and B. Chang, "PeerStreaming: Design and Implementation of an On-Demand Distributed Streaming System with DRM Capabilities," *Multimedia Systems Journal*, 2007.
- [22] S. Annapureddy, C. Gkantsidis, P. R. Rodriguez, and L. Massoulié, "Providing Video-on-Demand Using Peer-to-Peer Networks," *Microsoft Research Technical Report*, MSR-TR-2005-147, Oct. 2005.