Noname manuscript No.
(will be inserted by the editor)

Towards Capacity and Profit Optimization of Video-on-Demand Services in a Peer-Assisted IPTV Platform

Yih-Farn Chen, Yennun Huang, Rittwik Jana, Hongbo Jiang, Michael Rabinovich, Jeremy Rahe, Bin Wei, and Zhen Xiao

Received: date / Accepted: date

Abstract This paper studies the conditions under which peer-to-peer (P2P) technology may be beneficial in providing IPTV services over typical network architectures. It has three major contributions. First, we contrast two network models used to study the performance of such a system: a commonly used logical “Internet as a cloud” model and a “physical” model that reflects the characteristics of the underlying network. Specifically, we show that the cloud model overlooks important architectural aspects of the network and may drastically overstate the benefits of P2P technology by a factor of 3 or more. Second, we propose an algorithm called Zebra that pre-stripes content across multiple peers during idle hours to speed up P2P content delivery in an IPTV environment with limited upload bandwidth. We also perform simulations to measure Zebra’s effectiveness at reducing load on the content server during peek hours. Third, we provide a cost-benefit analysis of P2P video content delivery, focusing on the profit trade-offs for different pricing/incentive models rather than purely on capacity maximization. In particular, we find that under high volume of video demand, a P2P built-in incentive model performs better than any other model, while the conventional no-P2P model generates more profits when the request rate is low. The flat-reward model generally falls in between the usage-based model and the built-in model in terms of profitability except for low request rates. We also find that built-in and flat-reward models are more profitable than the usage-based model for a wide range of subscriber community sizes.

Keywords IPTV, P2P streaming, Content distribution network, FTTN, Video-on-Demand.

1 Introduction

Internet protocol TV (IPTV) promises to offer viewers an innovative set of choices and control over their TV content. Two major U.S. telecommunication companies, AT&T and Verizon, have invested significantly to replace the copper lines in their networks with fiber optic cables for delivering many IPTV channels to residential customers.

A viewer can receive IPTV videos in good quality if the available bandwidth satisfies the need of video encoding rate for the target resolution and frame rate. To provide sufficient bandwidth for IPTV services, Internet service providers use high speed xDSL networks to deliver video content to viewers’ set-top boxes. As an example, the AT&T architecture for the U-Verse IPTV service uses Fiber-to-the-Neighborhood (FTTN) Networks. Its architecture consists of a small number of national super head-ends (SHE) and a large number of local video hub offices (VHO). The super head-ends serve as the national content aggregation points for broadcast and video on demand encoding. The local video hub offices provide aggregation and storage of local content. Each video hub office serves as a Video-On-Demand (VoD) library and distributes video content through local access switches to the customers. We refer to this network hierarchy as the “physical” model throughout the paper. FTTN networks can provide 20-25Mbps bandwidth to each household, which

---

Yih-Farn Chen, Rittwik Jana, and Bin Wei are with AT&T Shannon Research Laboratory; Y. Huang is with the Institute for Information Industry, Taiwan; H. Jiang and M. Rabinovich are with Case Western Reserve University; J. Rahe is with U.C. Berkeley. Funding for Rahe’s research has been provided by AT&T Labs, the State of California under the MICRO program, and by the Toshiba Corporation. Z.Xiao is with Peking University, China. The contact author is Yennun Huang: yennunhuang@gmail.com.

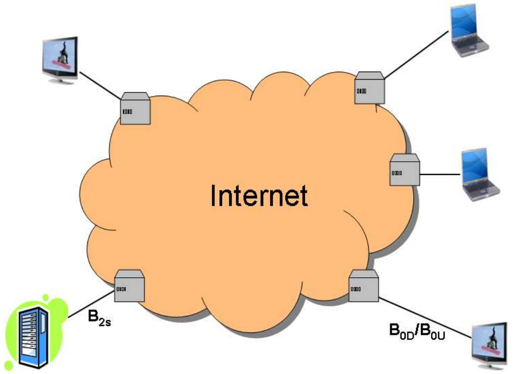

Fig. 1 Cloud Model

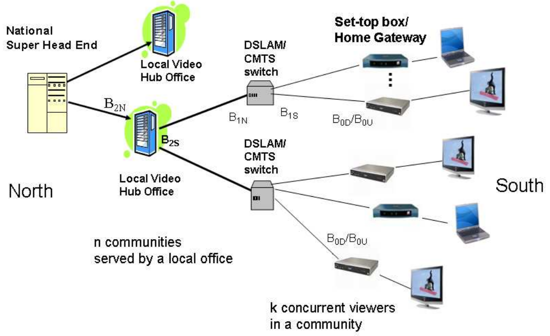

Fig. 2 Physical Model for IPTV Service

is typically enough to support several high quality TV streams as well as high speed Internet and Voice over IP (VoIP) services.

A potential challenge in providing IPTV services is to scale VoD delivery without incurring high deployment and maintenance cost. The video servers can quickly become a bottleneck as the number and frequency of VoD requests continue to rise. One solution to alleviate the load on servers is to use peer-to-peer (P2P) systems like BitTorrent [31] or Kontiki [19]. While early P2P systems were mostly used for file downloading, recently there have been several efforts on using the peer-to-peer approach to support live streaming [28][29][9][4][5][20] and VoD streaming[25][12][24][11]. Existing research studies that evaluate the benefits of P2P video content delivery typically do not consider the constraints of the underlying service infrastructure (e.g. [23][30]). Rather, they view the network as a “cloud”. Researchers, however, are increasingly aware of the need to reduce cross-ISP P2P traffic, while maintaining satisfactory P2P performance[6][32][17]. In this paper, we reveal the deficiency of this cloud model and investigate when P2P streaming can be beneficial in an IPTV environment. As we will see, P2P video sharing can be harmful under certain network conditions.

Another challenge for P2P streaming in an IPTV environment is the pricing strategy [8]. Most broadband ISPs today charge a flat fee for providing bandwidth. Usage-based pricing has emerged in some markets but even in those cases it is limited to volume-based pricing. Among the limited early work on pricing strategies for P2P, Adler, et al. [3] provided a comprehensive model applicable to a variety of P2P resource economies. Implementation of peer selection algorithms in realistic networking models like the IPTV environment was not addressed. Hefeeda et al. presented a cost-profit analysis of a P2P streaming service for heterogeneous peers with limited capacity [13]. The analysis shows that the service provider can achieve more profit by providing the appropriate incentives for participating peers. However, their analysis did not consider the bandwidth constraints of the underlying infrastructure and hence cannot be easily extended to our IPTV environment.

We make the following contributions in this paper:

- We compare two network models (the “cloud” model and the “physical” model) and show that the cloud model can dramatically overestimate P2P benefits.
- We propose an algorithm called Zebra that prestripes content across multiple peers during idle hours to speed up P2P content delivery in an IPTV environment with limited upload bandwidth. We also perform simulations to measure the effectiveness of the algorithm in offloading the content server during peak hours.
- We couple three P2P pricing models (flat-fee, usage-based, and built-in) with a “physical” model and study their trade-offs from a service provider’s perspective.

The rest of the paper is organized as follows. We describe the physical network model and constraints for the IPTV system in section 2. Section 2.2 provides the insights as to why a more accurate physical network model is necessary to realize a profitable IPTV system. Section 3 describes the design considerations behind the Zebra algorithm and the simulations that validate the effectiveness of the algorithm. Three different pricing models are analyzed and simulated in section 4. Section 5 discusses related work and Section 6 provides the conclusion.

2 Network Models

This section contrasts two network models that can be used in studying the performance of P2P video content delivery.

2.1 Cloud Model

Research in P2P streaming typically considers Internet at a logical level[23][30]: it represents the Internet at large as an abstract cloud and only considers the capacity of the content server and the characteristics of the access links to related hosts. We refer this view of the Internet as the "cloud model" as shown in Figure 1.

2.2 Physical Model

In contrast to the cloud model, the physical model considers the network architecture and bandwidth constraints of the underlying links and network devices. In [14], we described and analyzed the physical model of FTTN access networks for IPTV services. The model and analysis can also be applied to xDSL connections.

As shown in Figure 2, video streaming servers are organized in two levels - a local video hub office (VHO), which consists of a cluster of streaming servers or proxies to serve viewers directly, and national super head end (SHE) offices, which can distribute videos to local serving offices based on existing policies or on demand. We concentrate on video on demand (VoD) in this paper. Each local VHO office (often referred to as "local office" below) connects to a set of access switches such as FTTN switches through optical fiber cables. Each switch connects a *community* of IPTV service customers through twisted-pair copper wires or fibers. A community consists of all homes which are connected to the same access (xDSL) switch. A local VHO also includes a service router to connect to a national SHE office. These uplinks (or "north-bound links") of local offices are implemented over high-speed optical fiber networks.

The following parameters are used throughout the paper:

- B_{0D}: Download bandwidth into a home.
- B_{0U}: Upload bandwidth out of a home.
- B_{1S}: Total capacity of south-bound links (downlinks) of a local access switch.
- B_{1N}: Capacity of the north-bound link (uplink) of an access switch determined by the total bandwidth of north-bound fibers from a switch to a local VHO and the switching capacity of the service router in the VHO.
- B_{2S}: Maximum throughput in a local VHO determined by capacities of service routers, optical network cables and/or streaming servers in the VHO.
- u: Average streaming bit rate for a video.
- N_c: Maximum number of concurrent viewers supported by a local VHO.

As an example, the network for the U-Verse service allocates 20 to $25M$bps download bandwidth ($B_{0D} \leq 25M$bps) and $1M$bps upload bandwidth ($B_{0U} \leq 1M$bps) to each home. The AT&T network under U-Verse uses an FTTN switch which has a maximum of $24G$bps downlink (or "south-side") switching capacity ($B_{1S} \leq 24G$bps). Each FTTN switch can connect an OC-24 fiber to a service router in a local VHO ($B_{1N} \leq 1.244G$bps). The service router in a local VHO could then connect an OC-192 fiber to national SHE offices. Each high-definition (HD) channel uses 6 Mbps bandwidth and each standard-definition (SD) channel uses 2 Mbps bandwidth.

2.3 Network Constraints under Physical Model

In a physical network environment, all P2P upload traffic has to traverse the access switches and service routers that connect the peers. P2P streaming will increase the load of access switches, local offices and national offices; in particular, inter-neighborhood sharing creates traffic that traverses the link from the sending peer to the local VHO and then the link from the local VHO to the receiving peer.

Compared with the conventional IPTV services, P2P sharing within a community may not be beneficial if the south-bound link bandwidth of an access switch is the bottleneck. However, P2P sharing *within a community* decreases the load on the north-bound link of an access switch. Therefore, P2P sharing within a community will have the most benefit if the infrastructure bottleneck is on the north-bound link bandwidth of an access switch.

Similarly, P2P sharing among peers across communities increases the traffic on both the north-bound links and the south-bound links of access switches. If the network bottleneck is in either B_{1N} or B_{1S}, P2P sharing among peers in all communities creates more congestion for the switches and decreases the number of concurrent viewers which can be served by a local office. In this case, P2P sharing across communities is not beneficial for IPTV service providers. Also, if an IPTV service provider can apply content distribution network (CDN) technologies such as caching and replication to reduce the workload in SHE, the benefit of P2P sharing across communities in a VHO is very limited. The detailed analysis of network constraints for P2P IPTV services can be found in [14].

A key insight of this paper is that using the "cloud model" for P2P streaming is overly simplistic. More

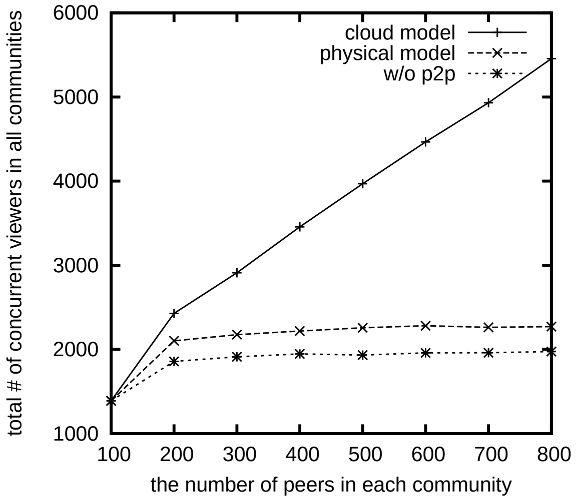

Fig. 3 Concurrent capacity vs. number of users

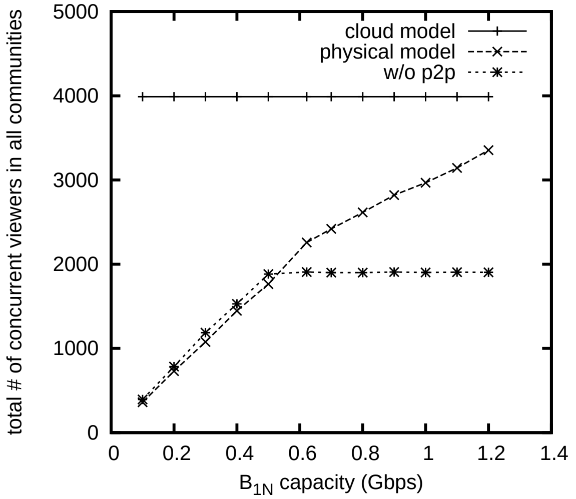

Fig. 4 Concurrent capacity vs. bandwidth of the office-to-access-switch link

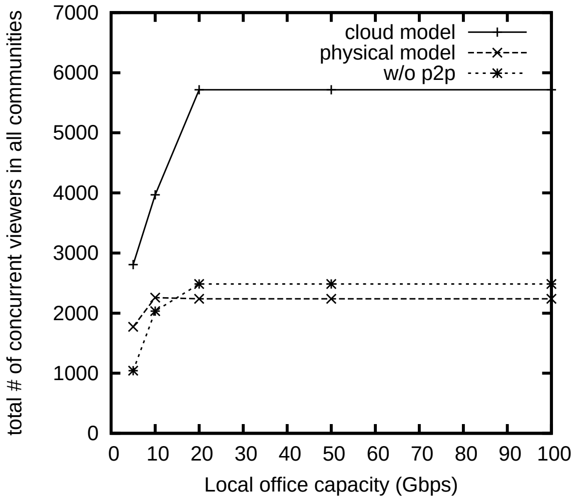

Fig. 5 Concurrent capacity vs. server capacity

realistic results can be obtained by considering the network at the physical infrastructure level. To demonstrate our point, consider the following simple P2P algorithm. The content server receives a request for a video, identifies candidate peers with that video and spare upload capacity, and selects a random set among them to collectively serve the video. If not enough candidates are available to serve the video at its encoding rate, the server tries to serve the remaining portion itself, or denies the request if it cannot.

We simulated the performance of the system under the two models. For the physical model, we used a slice of the infrastructure of Figure 2 corresponding to one local office with 20 communities and considered the situation where the content server in the local office distributes video content to the viewers in these communities. For the cloud model, we assume the same content server and viewers are connected via the Internet cloud. We assume the same behavior for every node in the community: an idle user (i.e. the user not viewing a stream already) requests a stream with probability of 2% every time tick. A time tick occurs every minute. A peer may download only one stream at a time. There are 1000 video programs available for viewing. When a peer issues a request, it selects a program according to Zipf's popularity distribution. The request will be rejected in case of insufficient server bandwidth and insufficient peer upload bandwidth. Each stream lasts 120 minutes and has a data rate of 6Mbps.[1] Once downloaded, the program remains available at the peer for a period called the stream time-to-live (stream TTL) with a default value of 1000 minutes. A peer may be turned off and on by its user. For simplicity, we assume that an operational peer is turned off with a fixed probability 0.1% on every time tick, and a non-operational peer is turned on with a fixed probability 0.5% on every tick. This means that on average every peer stays on five times longer than it stays off. This also means that an operational peer goes down at some point in any given 2-hour interval (our stream duration) with probability 11.4%. Note that these probabilities refer to the events of peers going up or down. Thus, they do not contradict previous studies showed that users only complete viewing a movie 10–20% of the time [1,2] (indeed, a settop box online status is independent of user viewing habits). We further assume that $B_{1N} = 0.622$ Gbps (OC-12), and $B_{2S} = 10$ Gbps. Each data point in the graphs throughout the paper is obtained by running the simulation program over 5000 time ticks and taking the average over the last 2500 time ticks (when the system reached a steady state in all the simulations).

---

[1] The HD stream encoding rate is constantly improving and we expect it to reach 6Mbps soon.

The results for the cloud and physical models are shown in Figure 345. The figures also include curves for the system that does not use P2P delivery under the physical model. Figure 3 shows the average number of concurrent viewers the system can support as the number of peers grows for fixed network and server capacities. The cloud model indicates that P2P delivery allows the system to serve more concurrent viewers and to scale to the growing number of viewers. However, the result is drastically different when the limitations of the physical infrastructure are brought into the picture. In fact, the cloud model could overestimate the benefit by mote than the factor of 2 when the number of peers in a community approachs 800 peers as shown in Figure 3. Not only does the P2P system serve fewer users, it does not scale with a growing number of users and has only a slight capacity advantage over the much simpler centralized delivery (which in fact turns to slight *disadvantage* for other parameter settings as seen in Figures 4 and 5). The reason behind this drastic change is the limitations of B_{1N}, the links between the local office and individual access switches. When P2P delivery occurs across different communities, two of these links are traversed: one upstream from the serving peer to the local office, and the other downstream from the local office to the receiving peer. Overall, these links are more heavily utilized under P2P delivery and more requests are denied.

Now consider the number of concurrent viewers under varying capacity of the office-to-access-switch link (Figure 4), when the community size is fixed at 500 viewers. The results for the cloud model are not affected by this link since the model does not consider it. However, the physical model reveals an important trend: the centralized delivery becomes quickly bottlenecked at the server and stops responding to the growing bandwidth of the office-to-access-switch link. On the other hand, with P2P delivery, improvement in this link's capacity produces a roughly linear growth in the number of concurrent viewers served, at least within the bandwidth range studied.

More differences are seen when we increase the server capacity instead (Figure 5). In this case, the cloud model quickly reaches the point where it serves all requested streams and stops being affected by the increase in server capacity. In particular, this result might indicate that it is highly beneficial to increase the server capacity from 10 Gbps to 20 Gbps. Under physical model, however, the number of concurrent viewers is unaffected by this change. Thus, the above investment would be useless under the simple algorithm we are considering. Comparing the P2P and centralized delivery under the physical model, the centralized delivery benefits from increased server capacity until it reaches 20 Gbps, after which the bottleneck shifts to the office-to-access-switch link. However, this bottleneck transpires later than in the P2P case. Overall, Figure 3 4 5 show that depending on whether or not the network operator plans to use P2P delivery, they should focus their investment on the office-to-access-switch link bandwidth or spread it between both server and office-to-access-switch link capacities. These trade-offs cannot be revealed under the conventional cloud model. Broadband speeds to homes are increasing. As fibre-to-the-home is deployed there will be an abundance of bandwidth (both uplink and downlink) in the future for p2p delivery. In a service provider market, we believe that the physical model will still hold true given the current technological and commercial trends.

3 Zebra - Scaling VoD Services with Pre-striped Content

As discussed in the previous section, a general P2P system does not scale well in a physical network (see figure 3). In this section, we present the design of Zebra, a P2P system that is designed to help scale VoD services in a typical IPTV architecture (such as the one from AT&T). Zebra exploits the application-specific properties of an IPTV-based VoD service to achieve optimizations not available to general P2P systems:

- Small number of peers (200-500) within the same local community
- Bounded latency between peers
- Frequent access to popular movies/TV shows
- Central server with a backup copy of all video files
- Access to peer (set-top box) availability characteristics
- Focus on meeting streaming video's bandwidth requirement rather than raw download speed
- Known peak and idle hours of VoD requests based on users' viewing habits and external events such as a local sports event

We elaborate on these application-specific properties and resulting design considerations in the following sections.

3.1 Zebra - Design Considerations

3.1.1 Insufficient Peer Bandwidth

One serious problem with using a conventional P2P solution, at least in the current technological landscape, is the limited upload bandwidth of each peer. As an example, AT&T architecture under the U-verse service allocates 1 Mbps for the upload bandwidth as default. If we limit the contribution of each peer to be no more than 50% of its upload bandwidth, then an HDTV request (at the rate of 6 Mbps) would require finding 12 peers that have already downloaded that movie, which is unlikely in a small neighborhood. Note that the upload bandwidth in broadband home environments is constantly improving, and we are beginning to see up to 4 Mbps upload bandwidth available in certain situations. In fact, right now ISPs tend to artificially limit the amount of upload bandwidth because they have no incentive to let their customers use that bandwidth for unrelated, bandwidth intensive applications, such as file sharing in BitTorrent-like systems. However, ISPs have the option of increasing the uplink bandwidth at the modem to allow faster intra-neighborhood communication for P2P VoD distribution, while still capping upload traffic to the Internet. Indeed, this step would be highly recommended for a viable P2P-based distribution system of HDTV content.

3.1.2 Prefetching and Striping

Since movies and TV shows are characterized by high degree of predictability in what will be watched, we can prefetch the most popular TV shows and movies to peers during off-peak hours at less than the speed necessary for live streaming.

In addition, we can break the pre-fetched movies into many small pieces that are evenly distributed (striped) throughout the P2P network. Even though each peer can only afford to contribute limited upload bandwidth, the aggregate is sufficient to support HDTV delivery.

For example, ten peers with pre-cached content can serve blocks at a steady rate of 200 Kbps (20% of the upload bandwidth) to satisfy an SDTV request from a peer, or 30 peers with 200 Kbps for HDTV delivery. The choice on the number of peers reflects a trade-off between the desire to avoid disruption to users' normal Internet experience and the management overhead. Spreading the required upload bandwidth over more peers results in less bandwidth demand on each peer, and hence reduces the potential degradation a user might experience during a normal Internet surfing session. It also increases the robustness of the system due to peer departure and failures. On the other hand, the management of a large number of peers incurs overhead, especially when the peer selection algorithm needs to run relatively frequently.

We can also modify the scheme to take into account that some movies are more popular than others; multiple copies of the stripe set may be used to ensure that one single group of peers is not responsible for handling all requests of a popular movie while another group remains idle waiting for a request of an unpopular movie.

3.1.3 Indexing and Lookup

In order to retrieve a desired file from a P2P system, the client must perform a lookup of some sort to determine which peer(s) has the file. To support this lookup, P2P systems either broadcast the search query by flooding or implement a distributed hash table (DHT) [22]. DHTs are designed to scale well to many nodes. However, we have a small number of nodes and could easily afford to keep track of which files each node has. In this initial design, we assume that the central server for the neighborhood receives every movie request and selects the set of peers to satisfy the movie request. In particular, the server has knowledge of the currently available spare upload capacity of every peer. In other words, due to the small, fixed membership in our network and the use of error-correction to obviate the need for shuffling block ownership, we have eliminated all lookup costs.

3.1.4 Profiling by Central Server

The central server maintains certain state information about the peers on the network. It keeps track of which peers are currently up, which peers stripe which movies, and the current spare upload capacity of each peer. The server uses this information to select a necessary set of peers to satisfy a given movie request. If a server peer goes down unexpectedly, the fixed inter-peer latency allows quick detection of this condition; the peer responds by issuing an update message and a high-priority block fetch to the central server.

Long-term historical information about which peers are up at which times in every FTTN neighborhood might be collected and used in tailoring the amount of error correction used in encoding pre-fetched movies.

3.1.5 Dealing with Peer Failures

As some peers may fail, redundancy is necessary to guarantee smooth delivery. Rather than storing entire extra copies, we can obtain fault tolerance by leveraging the fact that our movie data is being stored on many machines, rather than just one: before breaking the movie into pieces, we can use an error-correcting code (ECC), like an erasure code with a threshold (say 80%) [21]. The file can be recovered if any subset of 80% can be fetched. Such an encoding scheme has much less overhead than using entire copies.

With erasure coding, the movie file is broken into segments. Each segment is encoded as, say, 12 blocks, and any ten of which are sufficient to reconstruct the segment. Because segment reconstruction can occur only when 10 blocks are downloaded in their entirety, buffering is essential in our approach: a downloading peer delays movie rendering until the first segment is downloaded and reconstructed, and then downloads the blocks of the future segments while the current segment is being rendered. The buffering, and the initial viewing delay that stems from it, is also employed in the current Internet streaming delivery, except that today it is used mostly for jitter reduction. Companies like Vudu [33] preload the initial blocks of the most popular movies on their set-top boxes to give users instant access to these movies while downloading other blocks through their P2P network.

Because of buffering, when a downloading peer detects failure of a peer used for movie delivery, there is usually significant time left until the interrupted block will be needed for viewing. The downloading peer can use this time to contact the server and obtain the identity of an alternative peer for the interrupted block, or obtain this block from the central server directly.

We can use as much error-correction as is efficient from a storage/bandwidth standpoint, but since we have a central server that has all the data and can bridge the gap in the event that the P2P network "fails" to fetch the data, we do not need to be forced into using enough correction for the worst-case scenario.

3.1.6 Conceptual Similarities to RAID

In essence, we are distributing data across many nodes in order to get higher bandwidth and using error-correction to handle failed nodes. There is a strong analogy to a very large RAID array. However, it is more than just a disk array, because the individual peers also actively request data from each other. Rather, this system results also from the P2P design principle of scalability. The more nodes you add to a neighborhood, the higher the demand on the bottleneck link and the servers on the remote side. By letting the network itself shoulder some of that load, the system as a whole scales better.

3.1.7 Caching

Although we have so far focused on the striped prefetching system, peers also maintain a cache of movies they watch. This is necessary for simple things like rewinding, as well as allowing customers to rent movies (in which they can watch the movie as many times as they wish during a certain time period). Since the ISP knows which movies a customer has watched, it may be able to predict which movies the customer will watch in the future with the help of a recommendation system that will cater for customers with predictable viewing habits. Given the growing availability of low-cost storage, it is now feasible to prefetch entire movies to some customers' set-top boxes. This is particularly cheap since it can be multicast to many homes simultaneously during off-peak hours. We are currently investigating how the bandwidth savings vary depending on how storage is allocated between peer caches and striped pre-fetching.

3.1.8 Connection Overhead

Note that it may not be necessary to have a large number of open TCP connections (which might be a strain). Error-correction is already present at a higher level and the video software keeps its own re-order buffer, over a much larger window, so there is no point in TCP duplicating that. Furthermore, tight control from the central server over the P2P delivery process obviates the need for TCP congestion and flow control. The communication is more efficiently done over UDP.

3.2 Zebra Simulation Model

To study how Zebra system helps to reduce the server workload and increase the VoD service capacity, we use the simulation model which is similar to the one used in Section 2.2. We use the following peer failure model: every time-tick an operational peer goes down with probability 1% and a failed peer repairs with probability 5%.

We assume that once a day, the server decides on the set of movies to be made available and on the number of copies of each movie. We assume prefetching is done during off-hours when bandwidth is plentiful. We do not model the bandwidth consumption or content distribution in the prefetching process. For simplicity, we assume that only prefetched striped movies are used for peer-to-peer delivery. The movies acquired as the result of prior viewing are not used for this purpose.

The following parameters are used in describing the Zebra model (in addition to those specified in Section 2.2):

- Z_N: total number of videos in the VoD library
- Z_n: number of striped videos; the rest in the library is not striped
- Z_p: number of required serving peers for each copy of a striped video
- Z_c: number of copies of each striped video

- Z_r: rate of erasure coding; for example, a rate value of $\frac{5}{6}$ means $\frac{5}{6}$ of the total number of stripes is needed to recover the original video
- Z_s: number of stripes for each copy of the video: $Z_s = Z_p/Z_r$.
- Z_g: size of the peer storage (in GB) reserved for P2P content delivery; a 5GB storage on each peer in a 300-member community would allow roughly 800 SD movies at 1.5GB each (and striped with $Z_r = \frac{4}{5}$) to be stored in the local community.

We enumerate below some of the default parameter values used in the simulation model.

1. R_p: Request probability of 5% for a new download if a peer is inactive
2. Zipf's popularity distribution of videos in the VoD library with parameter 1.
3. N: Total number of subscribers per community = 300
4. Number of communities = 1
5. $Z_N = 500$
6. Server capacity = 500 Mbps
7. $B_{0D} = 20$ Mbps, $B_{0U} = 1$ Mbps
8. Stream TTL = 100 ticks

The movie popularity is modeled according to the Zipf's distribution with parapmeter 1. The resulting cumulative probablity function for 500 movies is shown in Figure 6. We observe that the top 50 movies (10%) accounts for 60% of the requests. The popularity distribution is dependent on the size of the movie database and the number of requests. We observe a conservative estimate for the popularity distribution of VoDs. We experiment on one community with 300 peers and a local VoD server capacity of 500 Mbps. We are mainly interested in understanding the amount of server capacity savings at the expense of using peer assistance within a community for a wide range of parameters.

In the following, we discuss some of the simulation results based on the above default parameter space. Any deviation from the default parameter space is detailed within each experiment. Movies are striped across peers at the beginning of each simulation. In experiments detailed below, the number of peers per stripe and the number of copies per movie are varied within a community. The average capacity/bandwidth of the server at steady state is computed over a simulation length of 5000 ticks. Normally, each peer provides a stripe at 200 Kbps for a length of 100 ticks. A requesting peer is therefore supplied by 10 peers in collection at a playout rate of 2 Mbps in a given tick. Using a peer-first approach, we exhaust peers' uplink bandwidth first before requesting the local server to supplement any remaining needs.

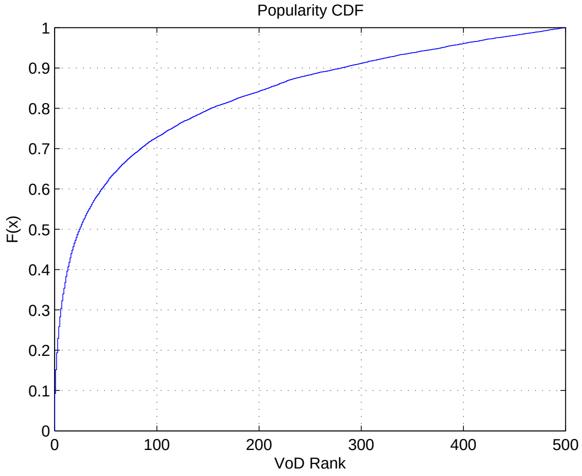

Fig. 6 VoD simulated popularity - Zipf distribution

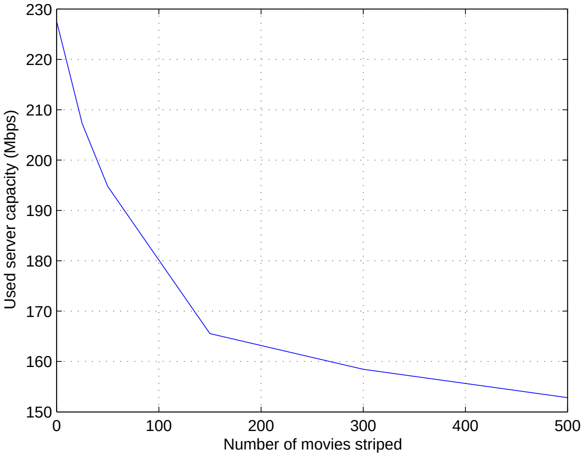

Fig. 7 Server bandwidth utilization (Mbps) vs number of striped movies among peers in the community Z_n: $Z_N = 500$, $Z_p = 10$, $Z_r = 5/6$, $Z_c = 5$, $R_p = 0.05$

Figure 7 plots the utilized server capacity vs the number of movies that are striped among the peers in the community. On the one extreme, no movies are striped ($Z_n = 0$) and hence the server is always utilized to supply any requests. On the other extreme, all movies are striped ($Z_n = Z_N$) and hence available at local peers. Only the top Z_n movies out of the 500 are striped and each movie is striped across 12 peers with a total of five copies per movie in the community. It is observed that as more movies are striped locally, the dependence on the server is reduced by about 11%. Note that the server capacity does not asymptotically approach zero, since not all peers can be available for uploading to requesting peers at a given point in time. Any overflow has to be accommodated by the local server.

Figure 8 shows the active number of peers downloading a movie as a function of simulation tick itera-

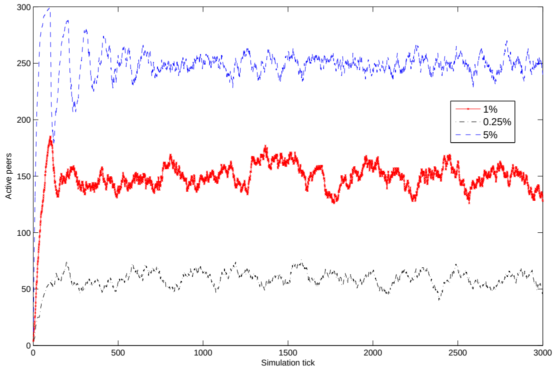

Fig. 8 Active viewing peers set vs simulation tick time: $Z_c = 1$, $R_p = 0.0025$, 0.01 and 0.05

tions. There is an initial ramp-up procedure after which it stabilizes to an average value. With increasing request rates (from 0.25% to 5%), the average number of active peers also grow. This simulation uses one copy per movie of the entire VoD database.

Figure 9 shows the average number of active users and the server bandwidth utilization vs varying request probability. The request probability is the chance of generating a new download if the peer is not downloading in a particular tick. As expected, the server utilization is seen to increase with request probability. The utilization of the server bandwidth fluctuates as popular VoD requests cannot be delivered by peers only. We use a peer-first policy where requests are accommodated by peers first, failing which the server is contacted for the remaining bandwidth needs. There is a sharp increase of about 28% in the server bandwidth utilization as request probability increases from 1% to 10%. Further increase in the request probability has a much lower rate of increase. Similarly, note that as the request probability increases from 0.25% to 20% the viewership asymptotes to the maximum community size of 300.

Figure 10 shows the average server bandwidth utilization as a function of increasing uplink bandwidth. As more uplink bandwidth is provisioned per peer there is very little use of server capacity. It is interesting to note that increasing uplink bandwidth by a factor of 2 (from 1 Mbps to 2 Mbps), results in a 25% drop in server utilization. Provisioning any further uplink bandwidth does not result in such a dramatic savings. This can be explained by the following simple calculations: Given that the average number of active peers is 250, $Z_s = 10$, and the serving peer is giving 200 Kbps to each receiving peer, in a 300-member community, the average bandwidth required from each peer to satisfy all peering activities is about 1.6 Mbps. The increase of uplink bandwidth would certainly be welcomed for HDTV content as each peer would have to contribute 600 Kpbs with $Z_s = 10$.

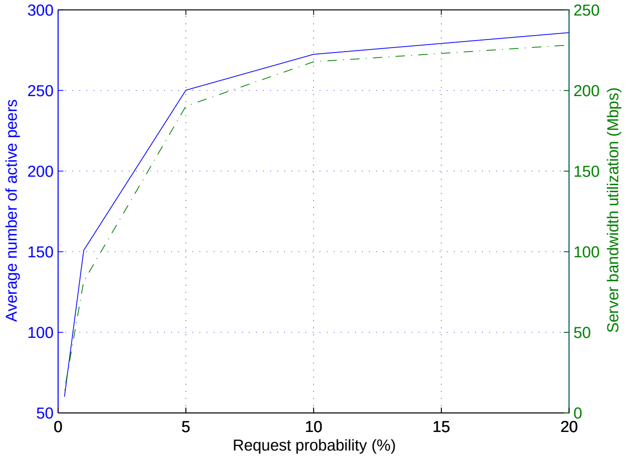

Fig. 9 Average number of active peers (left y-axis) and Server bandwidth utilization (right y-axis -.) vs Request probability

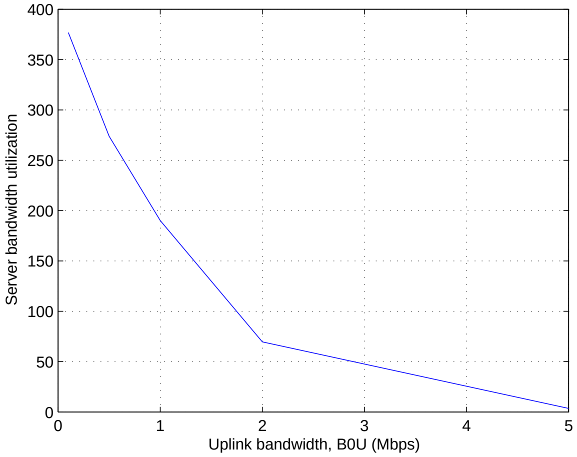

Fig. 10 Server bandwidth utilization (Mbps) vs increasing uplink b/w

Figure 11 shows the server bandwidth utilization as a function of the number of stripes per movie. This simulation used one copy per movie for the entire database. As the number of stripes increase, there is less dependency on the server. The aggregate playout rate is still kept at 2 Mbps, hence, as the number of peers per stripe increase, each peer uploads at a lower nominal rate. Z_r is kept fixed at a ratio of $\frac{5}{6}$.

Figure 12 shows the server bandwidth utilization for SD and HD streams as a function of time. The request probability was lowered to 1% to accommodate HD streams. SD streams were striped across 12 peers (requiring 10 peers to provide for 2 Mbps). HD streams were similarly striped across 12 peers (requiring 10 peers to provide for 6 Mbps). We observed that

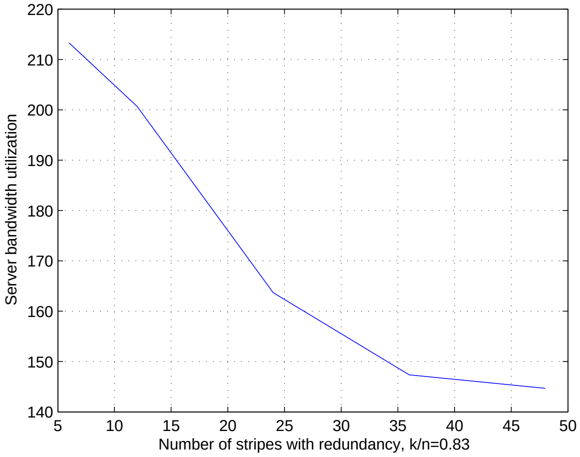

Fig. 11 Server bandwidth utilization (Mbps) vs number of stripes per chunk: $Z_r = \frac{5}{6}$, $Z_c = 1$

with a 500 Mbps server capacity, HD streams were sometimes interrupted due to exceeding server capacity. Average SD and HD server capacity utlizations were 96.4Mbps and 450.5Mbps respectively.

In our study so far, we have used a limited form of the Zebra algorithm in that only the top movies are striped with the same Z_p, Z_c, and Z_r. In its more general form, one would include a mapping table of videos of the library vs. peers in the neighborhood. Each video can have an arbitrary number of copies or stripes stored in any subset of the peers. The optimal peer assignment will be based on the available upload bandwidth from each peer, which depends on both the distance of the peer from the DSLAM switch and observed behavior of the upload activities from the peer.

4 Cost-Benefit Analysis

In order to encourage viewers to make their set-top boxes available for P2P sharing, some incentive may be given to peers who upload videos to other peers. This section analyzes the cost and benefit of deploying P2P technology on a physical network and compares its maximum possible profit to that of a conventional IPTV service.

4.1 Maximum Benefit for Conventional IPTV

Let r be the fee for a peer to download one movie and s be the average subscription fee paid by an IPTV subscriber, recalculated from the monthly period unit to the average movie duration. In this section, we will consider the average movie duration (120 minutes) as our time unit. For example, the \$99/month fee would be recalculated to be \$0.28 per time unit because there are on the average 360 2-hour slots in a month. For IPTV services, the maximum revenue in a local office per time unit is $R_{max} = rN_1 + sN_2$, where N_1 represents the number of concurrent viewers per time unit supported by a local office - with or without P2P incentives, and N_2 refers to the total number of subscribers supported by a local office. The maximum profit per time unit, P_{nop2p}, is

$$\begin{aligned} P_{nop2p} &= maximum\,income - IPTV\ expenses \\ &= rN_1 + sN_2 - E_{nop2p} \qquad (1) \end{aligned}$$

where rN_1 represents VoD income, sN_2 represents subscription income, and E_{nop2p} is the capital and operational expenses of the IPTV services per time unit.

4.2 P2P Incentive Models

To encourage P2P sharing among viewers, we consider three incentive models: Built-in model, Flat-reward model and Usage-based model.

4.2.1 Built-in Model

In this model, every set-top box includes P2P streaming software by default. Hence, P2P sharing is hidden from the viewers. The maximum profit per time unit is $P_b = rN_1 + sN_2 - E_{p2p}$, where E_{p2p} is the total operation and capital expenses per time unit for providing P2P IPTV services. It should be greater than E_{nop2p} because P2P software needs to be installed on servers and clients and hence will increase the cost of the infrastructure. Let's assume $E_{p2p} = E_{nop2p} + A_{p2p}$, where A_{p2p} includes the additional software license and maintenance fees paid for P2P software and additional hardware (such as disk storage). In the built-in model, we assume that the recurring software license and maintenance fees and the amortization of additional hardware result in each set-top box costing t dollars extra per time unit. Therefore, $A_{p2p} = tN_2$. Accordingly,

$$P_b = rN_1 + sN_2 - E_{nop2p} - tN_2 \qquad (2)$$

4.2.2 Flat-reward Model

In this model, a viewer signs up for the video sharing feature for a flat reward. Assume a fraction $w, (0 \leq w \leq 1)$ of viewers in a community sign up for video sharing and the reward is d dollars per time unit. The number of concurrent viewers per time unit supported by a local office is denoted to be N_1 and the total number of users in communities is denoted to be N_2 as before.

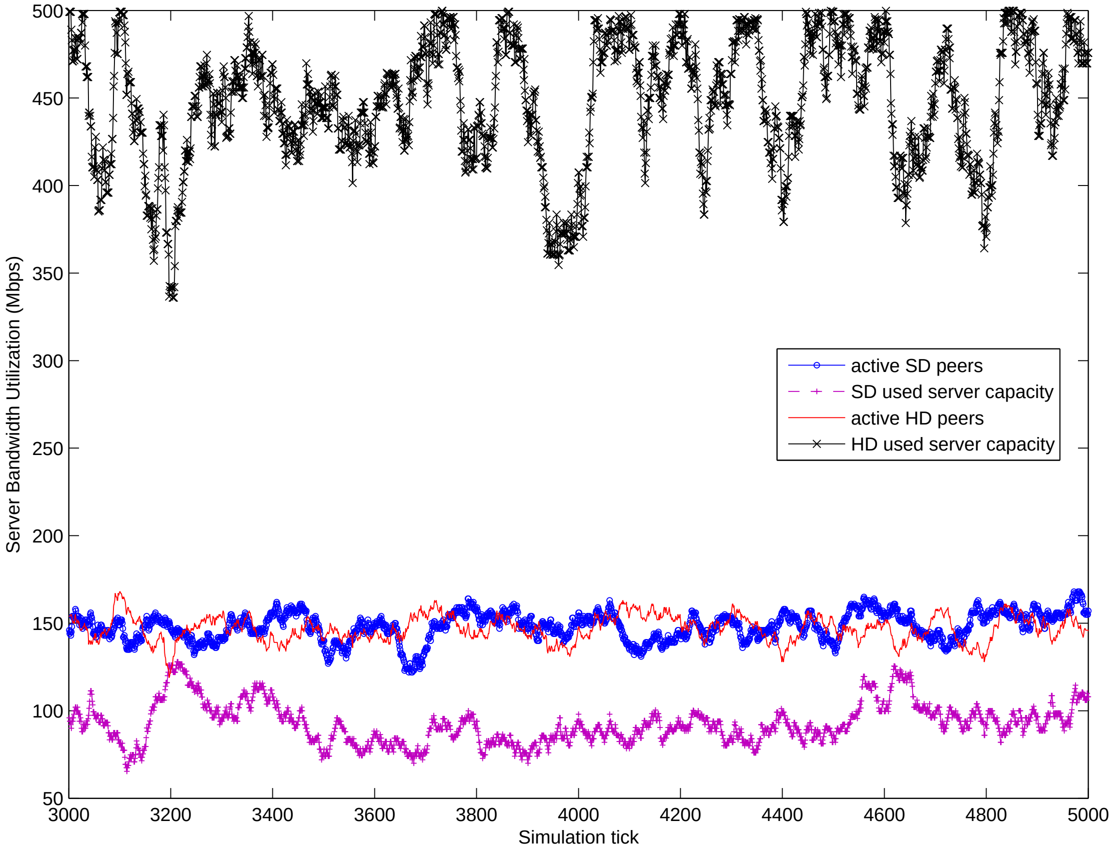

Fig. 12 Comparison of server utilization of SD and HD as a function of time: $Z_r = \frac{5}{6}$, $Z_c = 1$, $R_p = 0.01$

The maximum cost of incentive per time unit for the office is dwN_2. Each peer who signs up for the sharing needs to install and activate the P2P software on its set-top box. We assume that a service operator incurs the P2P software license fee only for those set-top boxes where P2P software is activated. Therefore, E_{p2p} equals $E_{nop2p} + twN_2$. The maximum profit per time unit in this model is

$$\begin{aligned} P_f &= total\ income - expenses - incentive \\ &= rN_1 + sN_2 - E_{p2p} - dwN_2 \\ &= rN_1 + sN_2 - E_{nop2p} - twN_2 - dwN_2 \qquad (3) \end{aligned}$$

In general, w depends on d: increasing d will increase the percentage of viewers willing to share videos and hence increase w. It should be noted that there is a trade-off to specify d here. On the one hand, increasing d could increase the number of concurrent viewers supported by a local office; i.e., it increases N_1 and thus could increase profit. On the other hand, increasing d will increase the cost to reward the users who sign up to allow P2P video sharing; i.e., it will increase dwN_2. We do not discuss how to achieve optimization in this work since a small change of system configurations may cause oscillation of the optimal point. Instead, we use a constant value for w in our experiments.

4.2.3 Usage-based model

In this model, a user who signs up for P2P sharing will get credit based on the number of bytes uploaded from his set-top box. Again, assume a fraction w of viewers in a community sign up for video sharing. Let q be the credit per bit uploaded from a viewer's set-top box. The number of bits uploaded from peers for P2P IPTV services in a time unit is u. The IPTV service provider gives incentives to these supporting peers based on their contributed bandwidth. In this model, the total reward given by an IPTV service provider to peers in a local office per time unit is qu. The maximum income per time unit in this model is

$$\begin{aligned} P_s &= rN_1 + sN_2 - E_{p2p} - qu \\ &= rN_1 + sN_2 - E_{nop2p} - twN_2 - qu \qquad (4) \end{aligned}$$

As an example, to compare the maximum profit per time unit under the conventional and the three incentive models, we assume that each viewer pays 3 dollars to watch a movie (r=3) and each movie lasts about 120 minutes. With download bandwidth B_{0D} of 22 Mbps, upload bandwidth B_{0U} of 1Mbps, and SDTV streaming rate of 2 Mbps, each SD movie consumes 14.4Gb or 1.8GB. Note that this would require either 2 streams of 1 Mbps, or in the case of Zebra, 10 streams of 200 Kbps if $Z_s = 10$, from peers for P2P delivery. If we do not consider the "push" cost of Zebra during idle hours, then the total payout is the same in either case in the usage-based model. We further assume that the capital/software/operational cost of each office is $1 million per year ($22.83 per time unit), the average monthly subscription fee is $99 ($0.229 per time unit), and the additional cost of incorporating P2P software and hardware (disk storage) on each set-top box per time unit is $0.1. We assume that B_{2S} = 50 Gbps. Note that B_{2S} is also constrained by the total streaming throughput from the server, which is about 10 Gbps.

We can now plot the profit per unit time comparing the conventional model and the incentive models of VoD services with varying B_{1S} (1-30Gbps) and B_{1N} (1-10Gbps) capacities, as shown in Figure 13. The maximum number of concurrent viewers are estimated according to a linear optimization program as discussed in [14]. In Figure 13, upper bounds for N_1 are used to illustrate the profit capacity surfaces. The profit number ramps up faster for the built-in model (smooth surface) (with a given B_{1N}) compared to the no-P2P model (striped surface) as we increase the bandwidth of B_{1S} until it reaches a plateau. Typical values of $w = 0.5$, $t = 0.1$, $q = 0.01$ per Gb, and $d = 0.05$ were used to estimate these capacities. The total number of subscribers, N_2 were 3000, made up of 10 communities with 300 peers per community.

4.3 Maximizing IPTV Service Provider Profit

To study the benefit of P2P technology for an IPTV service provider under various incentive models, we performed a simulation study using the Zebra algorithm. Based on the analysis in section 2, which shows that the benefit of P2P sharing among peers in different communities is very limited [14], we only consider P2P sharing within a community and simulate a system comprised of the local office and one community. We consider "peer-first" approach where peers are selected first if the requested video can be severed by peers, and local office will still have to serve the video if the requested video can not be severed by peers.

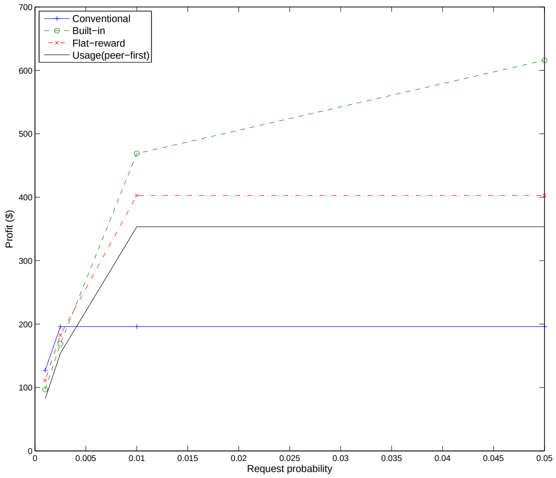

Fig. 14 Profit per unit time for the three incentive models vs request rate, 1 community, 300 peers/community, 500 movies, $Z_c = 1, r = 3, s = 0.229, t = 0.1, E_{nop2p} = 22.83, w = 0.5, d = 0.05$

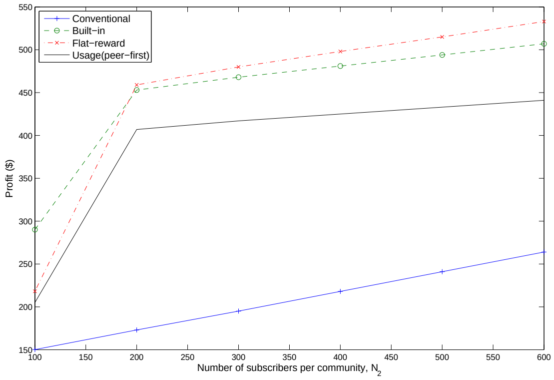

Fig. 15 Profit per unit time for the three incentive models vs number of peers, 1 community, 500 movies, $Z_c = 1, r = 3, s = 0.229, t = 0.1, E_{nop2p} = 22.83, w = 0.5, d = 0.05, R_p = 1\%$

We assume the same simulation model as described in section 2.2, using the physical network model. We assume that viewing each movie costs $3 (even if it is viewed from the local cache), peer incentive in the flat-reward model is $0.05 per time unit, and peer incentive in the usage-based model is $0.01 per 1Gb.

Figure 14 and 15 compare the profits among the three incentive models: built-in, usage-based, and flat-reward. Figure 14 shows how the profit changes with request rate. Intuitively, with more user requests, there is a greater opportunity for the service provider to make money (up to the capacity of the system). Indeed, the unit time profit for all three incentive models increases with request probability (albeit at a reduced rate). All models come very close in the profit numbers when the

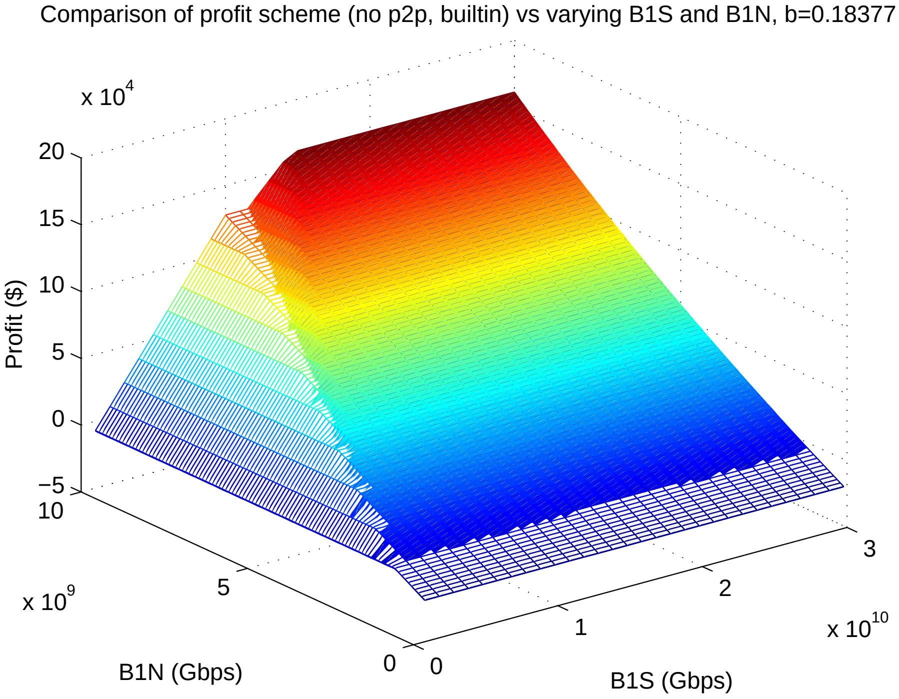

Fig. 13 Profit per unit time in the no-P2P model and built-in model under varying B_{1N} and B_{1S} capacities (this figure should be viewed in color)

request is low for a community of 300 subscribers. The conventional model yields more profit when the request rate is very low and it plateaus after it reaches its maximum capacity of 50 concurrent viewers. The usage-based model lags the built-in and flat-reward cases due to its additional payout once the request rate is over 0.5%. Under high request rates, the built-in model succeeds in using the aggregated P2P delivery capacity to support the most number of concurrent viewers (it peeks at adding 150 additional SDTV viewers if all 300 subscribers in a community with 1 Mbps upload limit all participate in P2P delivery). This can only be achieved when the content is pre-striped using an algorithm like Zebra to make sure that there are enough peers to support most movie requests. Traditional P2P algorithms do not guarantee enough peers with the requested content and sufficient bandwidth to satisfy the demand using only peers within local community.

Figure 15 shows how the profit changes with the number of subscribers at a fixed request probability ($R_p = 0.01$). Note that the linear growth of the conventional model comes purely from the increase in subscription revenue as it cannot add more concurrent viewers at this request rate. The built-in and flat-reward models have the best overall profit. Note that a usage-based model needs to compensate peers even when the server has spare capacity, while the cost/benefit of the built-in model depends on the utilization of the system needed to amortize its investment on the additional hardware and software. As we increase the number of subscribers, the flat-reward model continues to gain additional peers ($w = 0.5$)to help out with content delivery, but the built-in model does not enjoy as much profit due to the high deployment cost for 100% of the subscribers at this request rate.

In summary, as the number of peers increases, all P2P incentive models clearly generate more profit than the no-P2P model, because of the increased system capacity due to P2P content delivery. However, we see large differences among the incentive models as described above.

5 Related Work

Peer-assisted video-on-demand has been studied in [32] to provide users equal experience as traditional client-server approach. Their study found that P2P can reduce server bandwidth requirement significantly, especially when prefetching is used during network idle time. The work on how to improve VoD server efficiency can also be found at [16]. ISP friendly P2P techniques have been studied in [6, 18, 32] where they identify possibility to minimize cross-ISP traffic while still reaping most of the benefits with P2P. User-generated content in

YouTube has been analyzed in [7] to review viewing patterns, video life-cycles, request properties, content aliasing, etc. The potential of using P2P techniques for YouTube-like systems is also discussed.

The most related work is our previous conference paper [15], from which this journal paper work is evolved. The major additions are a new media management technique for P2P streaming and extended cost-benefit analysis. We first proposed the idea of striping media data across peers (Zebra) and using error correcting code to overcome transient failures at the AT&T Labs 2006 University Collaborations Symposium [8]. The authors in a recent paper[27] proposed a similar peer-to-peer system to push content to peers proactively and a load balancing algorithm to select serving peers in a distributed fashion. Their focus has however been to derive optimal placement strategies namely a full striping scheme and a code-based scheme that handle deterministic and stochastic demand models. Our proposed algorithm does not investigate data placement techniques, rather quantifies the operator-attainable profit using one such striping scheme. We also note that in a realistic setting, an operator may have full knowledge of each set-top box's viewing behavior. A combination of a recommendation system together with a suitable striping strategy may prove to be more useful.

One can conceive a P2P network as a large RAID array. The server keeps track of the allocation of media streams to peers and schedule them intelligently to provide the required sub-streams in real time. Incentives and micropayment for peer participation have also been discussed in the literature. The authors in [17] design a BitTorrent like protocol for P2P assisted VoD in IP-enabled set-top boxes. It leverages the storage capacity of those boxes for improved viewing experience and performs location aware content fetching to reduce cross-AS traffic on the Internet. Erasure codes [21] were used in [10] for peer assisted file delivery.

6 Conclusions

This paper studies the conditions under which P2P technology may be beneficial in providing IPTV services. We showed that the cloud model of the Internet frequently used in simulation studies of peer-to-peer systems may drastically overstate the benefits of P2P video content delivery. In particular, in a service provider model, P2P sharing across subscriber communities can increase the strain on the bottleneck elements of the access network infrastructure. Thus, one must consider physical network infrastructure to obtain more reliable results.

We proposed an algorithm called Zebra that pre-stripes content across multiple peers during idle hours to speed up P2P content delivery in an IPTV environment with limited upload bandwidth. Using simulation studies over a physical network model, we evaluated Zebra's behavior under various usage scenarios. In particular, we showed that it effectively offloads the VoD server. Thus, with our approach, the system with similar resources can support a significantly larger number of concurrent viewers in a community.

Finally, we provided a cost-benefit analysis for different pricing/incentive models. We considered three plausible incentive models: usage-based, where sub-scribers are compensated based on the bytes they upload to other subscribers; flat-reward, where a subscriber receives a fixed discount on the subscription fee if it agrees to participate in P2P delivery, and a built-in model, where every settop box comes to subscribers with embedded P2P functionality. We analyzeed the profitability tradeoffs among these models under different scenarios. While specific results will depend on the particular price and incentive values, our key finding is that the benefits of P2P delivery depend crucially on the pricing and incentive structure. Thus, these aspects must be considered along with capacity constraints in designing a platform for VoD delivery. In summary, P2P may not be beneficial for IPTV services unless we employ properly engineered algorithms and incentive strategies as discussed in this paper.

References

1. H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. *Understanding user behavior in large-scale video-on-demand systems*, pp. 333–344, Proc. of EuroSys 2006.
2. Either WWW05 or ICDCS05 paper. **TO BE FILLED BY ZHEN**
3. M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, D. Yao, *Optimal peer selection for p2p downloading and streaming*, Proc. of INFOCOM 2004.
4. S. Banerjee, Bobby Bhattacharjee, and C. Kommareddy, *Scalable application layer multicast*, in Proc. of ACM SIGCOMM 2002.
5. Mayank Bawa, Hrishikesh Deshpande, Hector Garica-Molina, *Transience of Peers and Streaming Media*, in ACM SIGCOMM Computer Communications Review, January, 2003.
6. R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, A. Zhang, *Improving Traffic Locality in BitTorrent via Biased Neighbor Selection*, in Proc. of ICDCS, 2006.
7. M. Cha, H. Kwak, P. Rodriguez, Y.Y. Ahn, and S. Moon, *I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System*, in Proc. of Internet Measurement Conference, October 2007.
8. Y.F. Chen, J. Rahe, A.J. Smith, Y. Huang, and B. Wei, *A P2P Distributed Storage Service for the AT&T IPTV Network*, in AT&T Labs 2006 University Collaborations Symposium, http://www.research.att.com/userfiles/File/U_Collaborations /2006_UnivCollab_Abstracts.htm.

9. H. Deshpande, M. Bawa and H. Garcia-Molina, *Streaming Live Media over a Peer-to-Peer Network*, Stanford database group technical report (2001-20), Aug. 2001.
10. Christos Gkantsidis and Pablo Rodriguez. Network Coding for Large Scale Content Distribution. In Infocom 2005.
11. Yang Guo, Kyoungwon Suh, Jim Kurose, and Don Towsley, *A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation*, in Proc. of IEEE Int. Conf. on Multimedia Expo (ICME'03), 2003.
12. Yang Guo, Kyoungwon Suh, Jim Kurose, Don Towsley *P2Cast: P2P Patching Scheme for VoD Service*, in Proc of 12th WWW, 2003.
13. M. Hefeeda and A. Habib and B. Bhargava, *Cost-profit analysis of a peer-to-peer media streaming architecture*, Technical report, CERIAS TR 2002-37, Purdue University, June 2003.
14. Y. Huang, Y. Chen, R. Jana, H. Jiang, M. Rabinovich, A. Reibman, B. Wei, and Z. Xiao, *Capacity Analysis of MediaGrid: a P2P IPTV Platform for Fiber to the Node Networks*, in IEEE Journal on Selected Areas in Communications special issue on Peer-to-Peer Communications and Applications, January 2007.
15. Y. Chen, Y. Huang, R. Jana, H. Jiang, M. Rabinovich, B. Wei, and Z. Xiao, *When is P2P technology beneficial to IPTV services?* the 17th NOSSDAV, June 2007.
16. Y. Choe, D. Schuff, J. Dyaberi, and V. Pai, *Improving VoD Server Efficiency with BitTorrent*, the 15th International Conference on Multimedia, September 2007.
17. V. Janardhan and H. Schulzrinne, *Peer assisted VoD for set-top box based IP network*, in Proc. of P2P-TV Workshop, August 2007.
18. T. Karagiannis and P. Rodriguez and K. Papagiannaki. *Should Internet Service Providers Fear Peer-Assisted Content Distribution?*, in Proc. the Internet Measurement Conference, 2005.
19. *Kontiki - www.kontiki.com*
20. X. Liao, H. Jin, Y. Liu, L. M. Ni, D. Deng, *AnySee: Peer-to-Peer Live Streaming*, in Proc. of IEEE INFOCOM, 2006.
21. Luigi Rizzo, *Effective erasure codes for reliable computer communication protocols*, Computer Communication Review, April 1997.
22. S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, *A Scalable Content-Addressable Network*, ACM SIGCOMM, August 2001.
23. V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, *Distributing Streaming Media Content Using Cooperative Networking*, ACM NOSSDAV, May 2002.
24. Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, *A measurement study of peer-to-peer file sharing systems*, in Proc. of ACM/SPIE on Multimedia Computing and Networking (MMCN'02), January 2002.
25. Simon Sheu, Kien A. Hua, Wallapak Tavanapong, *Chaining: A Generalized Batching Technique for Video-on-Demand Systems*, in Proc. of the IEEE Int'l Conf. On Multimedia Computing and System, June 1997.
26. *Skype -www.skype.com*
27. Kyoungwon Suh, Christophe Diot, Jim Kurose, Laurent Massoulie, Christoph Neumann, Don Towsley and Matteo Varvello, *Push-to-Peer Video-on-Demand System: Design and Evaluation*, in IEEE Journal on Selected Areas in Communication, Vol. 25, No. 9, December 2007.
28. Duc A. Tran, Kien A. Hua, Tai T. Do, *A Peer-to-Peer Architecture for Media Streaming*, in IEEE Journal on Selected Areas in Communication, Special Issue on Advances in Overlay Network, 2003.
29. Eveline Velos, Virgilio Almeida, Wagner Meira, Azer Bestavros, Shudong Jin, *A Hierarchical Characterization of a Live Streaming Media Workload*, in IEEE IMW'02, 2002.
30. Dongyan Xu and Sunil Suresh Kulkarni and Catherine Rosenberg and Heung-Keung, *A CDN-P2P Hybrid Architecture for Cost-Effective Streaming Media Distribution*, Computer Networks, Vol. 44, Issue.3, pp. 353-382, 2004.
31. *http://www.bittorrent.com*
32. Cheng Huang, Jin Li, Keith W. Ross, *Can Internet Video-On-Demand be Profitable*, SIGCOMM, August 27–31, 2007, Kyoto, Japan.
33. *http://vudulabs.com*