

# Architectural Choices for Video-on-Demand Systems

A. Al Hamra, E. W. Biersack, G. Urvoy-Keller  
Institut Eurécom  
B.P. 193, 06904 Sophia Antipolis, FRANCE  
email: {alhamra, erbi, urvoy}@eurecom.fr

## Abstract

Cost-effectiveness is of foremost importance for large scale VoD systems. We assume a VoD system where each video is split into two parts, the prefix and the suffix. We consider two new architectures: One architecture where the clients are equipped with set-top boxes that allow to store locally the prefix part of some/all popular videos and second architecture where the suffix is transmitted via satellite.

For each architecture, we develop a cost model to compute the delivery cost of videos. We show that these architectures are efficient and significantly reduce the system cost in many scenarios: (i) By more than 45% with set-top boxes at the client side, (ii) By more than 80% for satellite transmission of the suffix.

**Keywords:** content distribution network (CDN), overlay network, set-top box, satellite.

## 1 Introduction

Video-on-Demand (VoD) systems allow to support various applications such as distance learning, home entertainment, electronic commerce, to name but a few. However, the bandwidth-intensive nature of video calls for efficient and scalable architectures that serve many clients via a single multicast stream. Prior work on VoD can be classified into three categories:

- Open-loop systems [10, 2, 11]: In open-loop systems, the video is divided into many segments. Regardless the client requests, the server periodically and infinitely broadcasts segments each of which at its own rate.
- Closed-loop systems [7, 5, 4]: In closed-loop systems, clients contact the server to retrieve the video. Each request initiates a new unicast/multicast stream.
- Prefix caching assisted periodic broadcast [6, 3]: These systems combine open-loop and closed-loop approaches. This combination ensures a zero start-up delay and makes these systems suitable for both, popular and non popular videos.

In this paper, we present two new architectures to provide a scalable and efficient VoD service to a large client population. We assume that each video is split into two parts, the prefix and the suffix. In the first architecture, we provide clients with set-top boxes to store the prefix

part of some/all popular videos. In the second architecture, we transmit the suffix via satellite. For each architecture, we develop a cost model to compute the delivery cost of videos.

The rest of this paper is organized as follows: Section 2 presents related work. Section 3 describes the distribution network. In section 4, we derive the cost models for each of our architectures. Section 5 concludes the paper.

## 2 Contribution and Related Work

Providing a scalable and efficient VoD service to a wide client population has been extensively investigated in previous work. The basic idea to achieve scalability is to serve multiple clients via multicast.

Open-loop systems differ in the way they set the length and the transmission rate of each segment. Pyramid broadcasting [10] sets the same rate to all segments while the segment sizes follow a geometric series. Tailored transmission [2] sets the same length to all segments while the rate decreases as the segment number increases. You et al. [11] present an hybrid system that combines the two above methods.

While open-loop systems broadcast the video regardless of the request pattern of clients, closed-loop systems serve the video in response to clients' requests. With patching [7], the first client to arrive receives a dedicated stream from the server. A new client that arrives after the first one joins the initial unicast stream that is transformed into a multicast stream. At the same time, the new client receives a separate unicast stream for the part it missed from the initial stream. Gao et al. [5] extend this patching scheme with the inclusion of a threshold to reduce the cost of the unicast streams.

With the hierarchical merging system [4], when a new client arrives, the server initiates a unicast stream to that client. At the same time, the client listens to the closest (in time) stream (target) that is still active. When the client receives via unicast what it missed from the target stream, the initial unicast stream is terminated and the client listens only to the target stream, and the process repeats.

Guo et al. in [6], have developed a methodology to combine open-loop and closed-loop systems. They divide the video into two parts, the prefix and the suffix. The prefix is delivered via a closed-loop scheme while the suffix is multicast via an open-loop scheme.

Similarly, the PS model [3] combines both, open-loop and closed-loop systems. The PS model splits each video into a prefix and a suffix. The prefix is stored in prefix servers that can be placed at any level throughout the network other than the client side. The suffix is stored at the server, placed at the root of the network. The prefix servers send the prefix via *multicast controlled threshold* [5] while the server broadcasts the suffix via *tailored transmission* [2]. For more details on the PS model, please refer to [3].

In this paper we propose two new architectures for large scale VoD systems. We assume that each video is split into a prefix and a suffix. In the first architecture, we use the set-top box at the client side to store the prefix of some/all popular videos. In the second architecture, we transmit the suffix via satellite.

For each architecture, we derive from the PS model a cost model to compute the delivery cost of videos. In the cost models, we include not only the network transmission cost but also the *server* cost, which depends on both, the storage occupied and the number of input/output streams needed. We also account for the network transmission cost as a function of the number of clients that are simultaneously served by the multicast distribution (either from the prefix servers or the suffix server).

### 3 The distribution network

In our cost model, we assume that the topology of our distribution network is a  $m$ -ary tree with  $l$  levels (figure 1). A network tree model has many practical aspects. A tree model captures the hierarchical structure of a large-scale network, where large backbone routers service many smaller service providers which in turn service end-users. For example, a tree might include multiple levels, dividing a network into national, regional and local sub-networks.

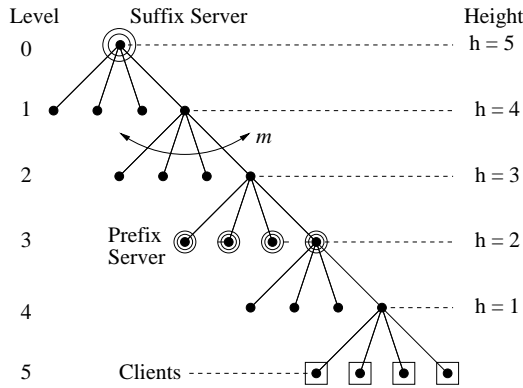


Figure 1: Video distribution network

The suffix server is assumed to be at the root of the tree. Prefix servers may be placed at any level of the distribution network other than the highest level (i.e. leaves). The clients are lumped together at the  $m^l$  leaf nodes. The number of clients watching simultaneously a video is not limited to  $m^l$  since a leaf node does not represent a single client but multiple clients that are for instance in the same

building. In this paper, we assume homogeneous client populations.

## 4 Evaluation of Architectural Choices

### 4.1 Analytical Model

In this section, we present the basic cost model and then derive the cost term for each architecture. We divide the cost of a VoD system into network and server cost. The network cost is proportional to the amount of network bandwidth needed for the transmission of the prefix and the suffix. The server cost depends on the disk storage used and the total number of input/output streams needed for both, the suffix server and the prefix servers. The total cost of the system can be computed as the sum of the total network and total server cost:

$$C^{system} = C_{netw}^{system} + \gamma C_{server}^{system} \quad (1)$$

To relate the network and the server cost, a normalization factor  $\gamma$  is introduced that allows us to explore various scenarios for the cost of the servers as compared to the cost of the transmission bandwidth.

The terms for the network and server cost are given by:

$$\begin{aligned} C_{netw}^{system} &= C_{netw}^{prefix} + C_{netw}^{suffix} \\ C_{server}^{system} &= C_{server}^{prefix} + C_{server}^{suffix} \end{aligned}$$

The server cost depends on both, the required amount of storage  $C_{sto}$  (in Megabit) and the amount of disk I/O bandwidth  $C_{I/O}$  (in Megabit/sec).

$$\begin{aligned} C_{server}^{prefix} &= \max(C_{I/O}^{prefix}, \beta C_{sto}^{prefix}) \\ C_{server}^{suffix} &= \max(C_{I/O}^{suffix}, \beta C_{sto}^{suffix}) \end{aligned}$$

To relate the cost for storage and for I/O, we introduce the normalization factor  $\beta$  that is determined as follows: If our server has a storage capacity of  $d_{sto}$  [Megabit] and an I/O bandwidth of  $d_{I/O}$  [Megabit/sec], then  $\beta = \frac{d_{I/O}}{d_{sto}}$ . Since the server will be either I/O limited or storage limited, the server cost is given as the *maximum* of  $C_{I/O}$  and  $\beta C_{sto}$ .

To model the case where the cost of the “last-hop link” towards the clients is not the same as the cost of the other links, we can set the cost for the last link to the clients ( $lhc$ ) to a value different from the cost for the other links.

This basic cost model has quite a few parameters. We present results only for a limited subset of parameter values that provide new insights. We will vary only the parameters  $\gamma$  and the last-hop cost  $lhc$ . We consider a distribution network with an out-degree  $m = 4$  and a number of levels  $l = 5$ . We expect that such a topology is representative for a wide distribution system that covers a large geographical areas of the size of a country such as France or the UK. If one wants to model a densely populated metropolitan area such as New York, one would choose  $l < 5$  (e.g.  $l = 2, 3$ ) and  $m > 4$  (e.g.  $m = 10$ ). The other parameters are chosen as follows: For the disk I/O

cost to disk storage cost ratio  $\beta$ , we choose  $\beta = 0.001$  (a realistic value for the current disk technology such as the IBM Ultrastar 72ZX disk). The video length is  $L = 90$  minutes.

## 4.2 Set-top Box at the Client Side for Prefix Storage

Today, set-top boxes at the client side provide a large amount of storage capacity at a low cost. For instance, the digital video recorder developed by Tivo [9] allows to store between 20 and 60 hours of MPEG II coded video and can receive transmissions at high data-rates. In this subsection, we present a new distribution architecture (called **P-hybrid**) that uses the set-top boxes to store video prefixes.

We derive the P-hybrid model from the PS model [3]. Both models use the same protocols to distribute the prefix and the suffix. However, in contrast to the PS model, the P-hybrid model allows the prefix to be stored not only at the prefix servers but also at the set-top boxes. In the P-hybrid model, when the prefix is stored at the prefix servers, the prefix is delivered to clients via controlled multicast as with the PS model. In this case, the cost of the prefix is the same for both models the P-hybrid and the PS ( $C_{P\text{-}hybrid}^{prefix} = C_{PS}^{prefix}$ ).

The prefix can also be downloaded directly to the set-top boxes. In this case, the prefix cost with the P-hybrid model, as compared to the PS model, is limited to the download cost of the prefix to the set-top box.

Both, the PS and the P-hybrid models store the suffix at the central server and deliver it to clients via tailored transmission. Thus, both models have the same suffix cost ( $C_{P\text{-}hybrid}^{suffix} = C_{PS}^{suffix}$ ).

### 4.2.1 Analytical Model

To compute the cost of the P-hybrid model, we partition the set of videos into two disjoint subsets, namely  $S_1$  and  $S_2$ , with  $S_1 \cap S_2 = \emptyset$ .  $S_1$  represents the set of videos whose prefix is stored in the **prefix servers**.  $S_2$  represents the set of videos whose prefix is stored in the **set-top boxes**. We calculate separately the cost for the videos in  $S_1$  and  $S_2$ . The total P-hybrid system cost is the sum of the system cost over all videos in the two subsets:

$$C_{P\text{-}hybrid}^{system} = C_{P\text{-}hybrid}^1(S_1) + C_{P\text{-}hybrid}^2(S_2).$$

For each video  $i$  in  $S_1$ , the P-hybrid model delivers the prefix and the suffix via the same protocols as the PS model does. Therefore, for each video  $i$  in  $S_1$ , the system cost can be computed using the PS model and the cost of  $S_1$  is:

$$C_{P\text{-}hybrid}^1(S_1) = \sum_{i \in S_1} C_{P\text{-}hybrid}^{system}(i) = \sum_{i \in S_1} C_{PS}^{system}(i)$$

$$\text{with } C_{PS}^{system}(i) = C_{PS}^{prefix}(i) + C_{PS}^{suffix}(i)$$

where  $C_{PS}^{prefix}(i)$  and  $C_{PS}^{suffix}(i)$  are given respectively in tables 1 and 2 in [3].

Concerning  $S_2$ , the **suffix cost** of each video  $i \in S_2$  is computed as in the case of the PS model since in both architectures, the suffix is delivered to the clients via tailored transmission. In contrast, the prefix cost comprises only the download cost of the prefix.

$$\begin{aligned} C_{P\text{-}hybrid}^2(S_2) &= \sum_{i \in S_2} C_{P\text{-}hybrid}^{prefix}(i) + \sum_{i \in S_2} C_{P\text{-}hybrid}^{suffix}(i) \\ &= \sum_{i \in S_2} C_{P\text{-}hybrid}^{prefix}(i) + \sum_{i \in S_2} C_{PS}^{suffix}(i), \end{aligned}$$

with

$$C_{P\text{-}hybrid}^{prefix}(i) = b \frac{D_i}{T_s} \sum_{j=1}^l m^j = b \frac{D_i}{T_s} \left( \frac{m^{l+1} - m}{m - 1} \right) \quad (2)$$

In equation (2),  $b$  is the playback rate of the video,  $D_i$  is the length of the prefix of video  $i$ , and  $T_s$  is the **download interval** (time between two consecutive downloads) of the prefix to the set-top box. The term  $\sum_{j=1}^l m^j$  accounts for the number of links traversed by the data at all levels during the download of the prefix. To minimize the cost of  $S_2$ , we must find the optimal set of prefix lengths  $\{D_i\}_{i \in S_2}$  of the videos in  $S_2$ . For this purpose, we solve the following problem:

$$\begin{aligned} \min_{i \in S_2} f &= C_{P\text{-}hybrid}^2(S_2) \\ \text{s.t. } \sum_{i \in S_2} D_i &\leq Cap \end{aligned}$$

Where  $Cap$  is the storage capacity of the set-top box. The problem expressed above is a non-linear programming problem subject to linear inequality constraints. To obtain a solution, we apply the `fmincon` package of Matlab.

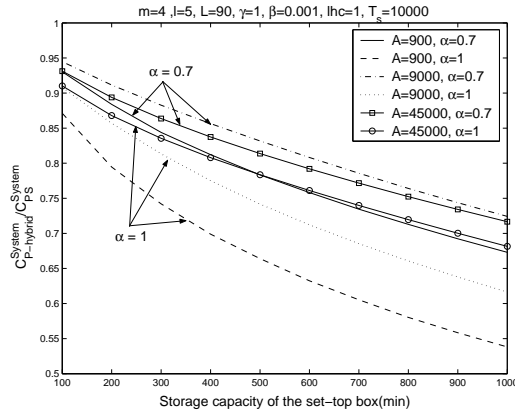
For a given partition of the videos between the two disjoint subsets  $S_1$  and  $S_2$ , we apply the PS model to compute  $C_{PS}^1(S_1)$  and we apply the `fmincon` package of Matlab to compute  $C_{P\text{-}hybrid}^2(S_2)$ . The P-hybrid system cost is the sum of  $S_1$  and  $S_2$ . However, to find the optimal total system cost, we must find which video prefixes should be stored in the set-top box. To do so, we present the following heuristic algorithm to find the near optimal split of the set of  $K$  videos between  $S_1$  and  $S_2$ . We sort the videos in decreasing order of popularity ( $N_i > N_j$  if  $i < j$ ) where  $N$  is the average number of simultaneous clients). We start with the case where all videos are in  $S_1$  (all the prefixes are stored at the prefix servers). In this case, the P-hybrid model is equivalent to the PS model.

In  $S_1$ , the most popular video consumes the largest fraction of the system resources among all videos. Thereby, we move the videos from  $S_1$  to  $S_2$ , one after the other, starting with the most popular one in the system. At each step, we compute the new cost of  $S_1$  ( $C_{PS}^{1\text{-}new}(S_1)$ ) and  $S_2$  ( $C_{P\text{-}hybrid}^{2\text{-}new}(S_2)$ ). We then compare the P-hybrid system cost at the current step  $C_{P\text{-}hybrid}^{system\text{-}new}$  and at the previous step  $C_{P\text{-}hybrid}^{system\text{-}opt}$ . If  $C_{P\text{-}hybrid}^{system\text{-}new} \geq C_{P\text{-}hybrid}^{system\text{-}opt}$  then we stop and the parameter values computed at the previous step are chosen.

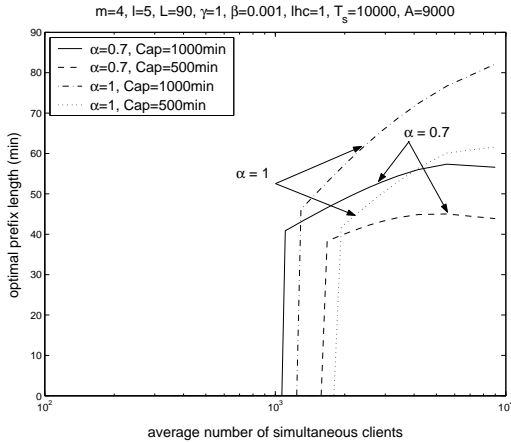
## 4.2.2 Results

To obtain insights about the advantages of having a set-top box at the client side, we plot in figure 2(a) the cost ratio of the P-hybrid to the PS models for  $\gamma = 1$  and homogeneous link cost (i.e.  $lhc = 1$ ). Indeed, we evaluated the P-hybrid model for other scenarios such as  $\gamma = \{0, 0.1\}$  and  $lhc = 0.1$ , and we found similar results. We set the download interval of the prefixes to  $T_s = 10000$  minutes (about *one week*) and the number of videos to  $K = 100$ . We vary the storage capacity  $Cap$  of the set top-box from 100 to 1000 minutes of video. The popularities of the videos are Zipf distributed with  $N_i = \frac{A}{i^\alpha}$  ( $N_i$  represents the average number of clients simultaneously viewing video  $i$ ).

As we can observe from figure 2(a), as the storage capacity  $Cap$  increases, the P-hybrid system cost reduces as compared to the PS system cost. This reduction exceeds



(a)  $C_{P-hybrid}^{system} / C_{PS}^{system}$



(b) Optimal prefix length in the set-up box

Figure 2:  $C_{P-hybrid}^{system} / C_{PS}^{system}$  and optimal prefix length in the set-up box for  $\gamma = 1$  and ( $lhc = 1$ ).

45% provided a storage capacity  $Cap = 1000$  minutes and a system with very few popular videos. Figure 2(b) shows

the optimal partitioning of the set-top box amongst videos for different values of  $A$ ,  $\alpha$  and  $Cap$ . The larger the buffer space at the client, the larger the number of videos that share that buffer and the longer the length of the prefixes stored locally. It might seem surprising that the length of the prefix does not necessarily increase monotonically with the popularity  $N$  of the video. In fact, the open-loop scheme (suffix transmission) performs well for very popular videos. Thus, it might be optimal to reduce the prefix length of the most popular video in order to free a place for the other popular ones.

Figure 2(b) also shows that, for given values of the parameter  $A$  and  $Cap$ , as  $\alpha$  increases, the number of videos that have their prefix stored at the client side decreases while the length of the prefix becomes longer. Indeed, for a given Zipf distribution ( $N_i = \frac{A}{i^\alpha}$ ), the popularity of video  $i$  decreases as  $\alpha$  increases. As we mentioned before, the P-hybrid model reduces the system cost as compared to the PS model by storing locally the prefix of the most popular videos. In contrast to  $\alpha = 0.7$ , when  $\alpha = 1$ , there are fewer popular videos in the system that should have their prefix stored in the set-top box.

Figure 2(a) also shows that for a given value of  $A$ , increasing  $\alpha$  increases the efficiency of the P-hybrid model. Actually, the P-hybrid model becomes more cost efficient as the cost of the most popular videos increases relative to the total system cost, which is the case when  $\alpha$  increases.

## 4.3 Use of Satellite for Suffix Transmission

Satellites are a very cost effective transmission medium for sending data to a large group of users. The cost of 1 Mbit/month satellite transmission bandwidth is about \$ 10,000 [8] whereas the cost for 1 Mbit/month terrestrial transmission bandwidth is \$1300 for 1 Mbit/sec during one month in case of a T1 line and \$350 for 1 Mbit/sec during one month in case of a OC-48 transmission link [1]. We now consider the case where the *suffix* is transmitted via satellite directly to the clients, while the prefix is transmitted from the prefix servers to the clients via the Internet. We refer to this distribution architecture as the **S-sat** model. In both the S-sat and the PS models, the prefix is stored at the prefix servers and delivered to clients via controlled multicast. As a result, the prefix cost is the same in both models ( $C_{PS}^{prefix} = C_{S-sat}^{prefix}$ ). On the other hand, both models schedule the suffix via tailored transmission. However, in contrast to the PS model, the S-sat model transmits the suffix via satellite instead of a terrestrial network. As a consequence, the cost term for  $C_{net}^{suffix}$  for the S-sat model is  $C_{net}^{suffix} = \sigma \cdot R_t^{min}$ , where  $R_t^{min}$  is the total server bandwidth needed to schedule the suffix via tailored transmission and  $\sigma$  is a weight factor that allows to express the cost for the satellite transmission in *relative* terms with respect to the other cost elements such as terrestrial transmission or server storage and I/O. The I/O and storage cost for the suffix remain the same in both models.

In the following, we will use two different values for  $\sigma$  namely  $\sigma = 100$  and  $\sigma = 500$ . In the light of the absolute prices given above, we consider both values as “conservative” in the sense that they are likely to overestimate the cost of a satellite transmission compared to a terrestrial transmission.

#### 4.3.1 Results

We see in figure 3(b) that for the S-sat model, the prefix decreases more rapidly with increasing number of clients  $N$  since the transmission of the suffix via satellite is less expensive compared to a transmission over a terrestrial network. The smaller the value of  $\sigma$ , the cheaper the satellite transmission and the more cost effective the S-sat model. For  $N > 10^2$ , the S-sat model is very cost effective (figure 3(a)). For a very high number of simultaneous clients, the suffix becomes eventually as large as possible (89 minutes)<sup>1</sup> and satellite suffix transmission can reduce the cost by up to 80% (for  $\gamma = 1, lhc = 1, \sigma = 100$ ). The cost reduction obviously depends on  $\sigma$ . For the case  $\sigma = 500$ ,  $\gamma = 1$ , and  $lhc = 0.1$ , the satellite transmission is quite expensive compared to a terrestrial transmission and as a result, the suffix satellite transmission is not competitive.  $\gamma = 0.1$  (figure 4(a)) makes the prefix servers cheaper, which allows to use more of them (equation (1), page 2). Such a cost reduction of the prefix servers benefits both the PS and the S-sat models. As a consequence, for  $\gamma = 0.1$ , the satellite transmission still remains, for large values of  $N$ , much more cost effective than the transmission of the suffix via terrestrial links.

If we completely ignore the server cost ( $\gamma = 0$ ) and the last hop cost is reduced ( $lhc = 0.1$ ), suffix transmission via satellite will remain more cost effective provided that satellite transmission is cheap ( $\sigma = 100$ , figure 4(b)).

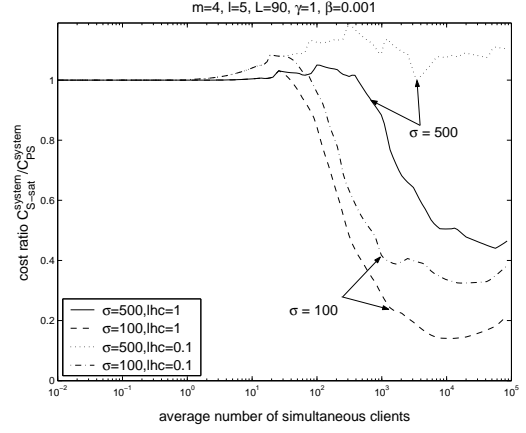
We eventually outline that our results showed (figure not shown) that using satellite suffix distribution as compared to terrestrial links has little impact on the placement of the prefix servers.

## 5 Conclusion and Future Work

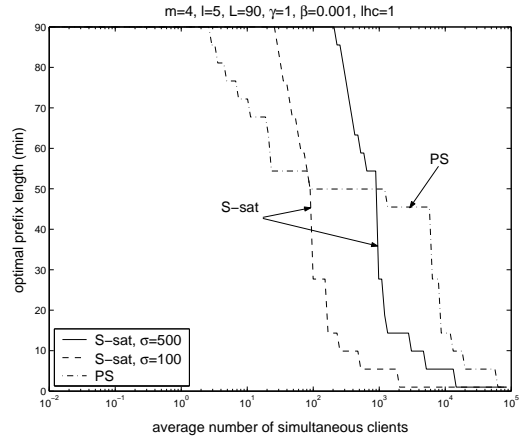
We have presented two new architectures for large scale VoD systems. We assumed that each video is split into two parts, the prefix and the suffix. In the first architecture, we allow clients to store locally the prefix of some/all popular videos. In the second one, we transmit the suffix via satellite. For each architecture, we developed a cost model to compute the delivery cost of videos. We applied these architectures to the PS model and we evaluated the overall reduction in the system cost. Our results showed that,

- Storing the prefixes of the most popular videos in the system at the client side can reduce efficiently the system cost by 30-45%.
- When the cost for the satellite transmission is low relative to the cost for the terrestrial transmission, using

<sup>1</sup>We limit the minimal length of the prefix to 1 minute in order to provide a zero start-up delay service.



(a)  $C_{S-sat}^{system} / C_{PS}^{system}$ ,  $\gamma = 1$  and  $lhc = \{1, 0.1\}$

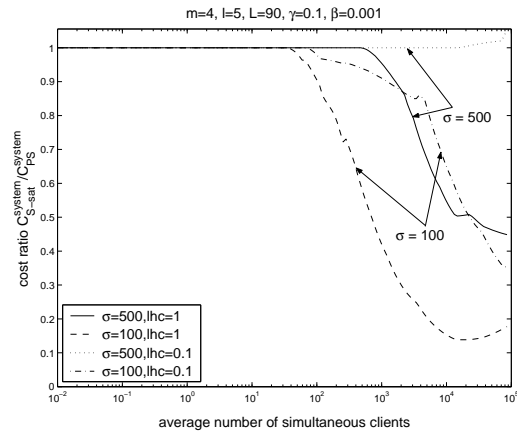


(b) optimal prefix length,  $\gamma = 1$  and  $lhc = 1$

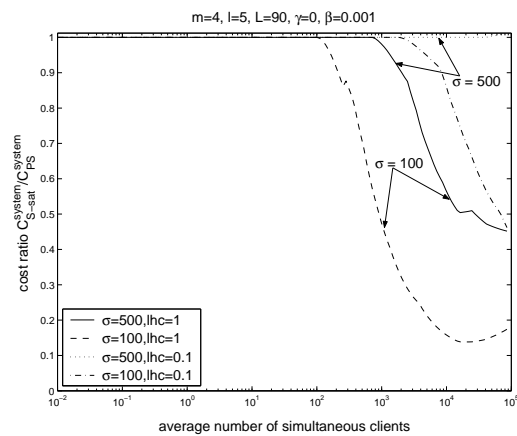
Figure 3: Cost ratio ( $C_{S-sat}^{system} / C_{PS}^{system}$ ) for  $\gamma = 1, lhc = \{1, 0.1\}$  and  $\sigma = \{100, 500\}$  and optimal prefix length for both, PS and S-sat models for  $\gamma = 1, (lhc) = 1$ , and  $\sigma = \{500, 100\}$  as a function of the number of clients  $N$ .

satellite to transmit the suffix can reduce the system cost by up to 80%.

In this work, we have assumed that (i) client requests for the same video are homogeneously distributed among all clients and (ii) the current video distribution network has a very regular structure with all clients being at the same distance from the root. A natural extension of this work would be to introduce heterogeneity in the video popularity and in the network construction. As a future work, we intend to study the impact of these extensions on the two architectures that we introduced here. While these extensions will clearly change the absolute values that we presented, we do not expect that they will change the broad conclusions that we obtained for both architectures.



(a)  $\gamma = 0.1$



(b)  $\gamma = 0$

Figure 4: Cost ratio ( $C_{S-sat}^{system} / C_{PS}^{system}$ ) for  $\gamma = \{0.1, 0\}$ , ( $lhc = \{1, 0.1\}$ ), and  $\sigma = \{100, 500\}$ .

## References

- [1] S. Banerjee et al., “Rich Media from the Masses”, HPL-2002-63R1, HP Lab, May 2002.
- [2] Y. Birk and R. Mondri, “Tailored Transmissions for efficient Near-Video-on-Demand Service”, In *Proc. of ICMCS*, pp. 226–231, June 1999.
- [3] D. Choi, E. W. Biersack, and G. Urvoy-Keller, “Cost-optimal Dimensioning of a Large Scale Video on Demand System”, In *Proc. of NGC*, October 2002.
- [4] D. Eager, M. Vernon, and J. Zahorjan, “Optimal and Efficient Merging Schedules for Video-on-Demand Servers”, In *Proc. 7th ACM Multimedia*, November 1999.
- [5] L. Gao and D. Towsley, “Threshold-Based Multicast for Continuous Media Delivery”, *IEEE Transactions on Multimedia*, 3(4):405–414, December 2001.
- [6] Y. Guo, S. Sen, and D. Towsley, “Prefix Caching assisted Periodic Broadcast: Framework and Techniques for Streaming Popular Videos”, In *Proc. of IEEE ICC 2002*, April 2002.
- [7] K. A. Hua, Y. Cai, and S. Sheu, “Patching : A Multicast Technique for True Video-on-Demand Services”, In *ACM Multimedia*, pp. 191–200, 1998.
- [8] J. Nonnenmacher, “Personal communication”, October 2002.
- [9] TiVo, “What is TiVo: Technical Aspects”, 2003.
- [10] S. Viswanathan and T. Imielinski, “Pyramid Broadcasting for Video On Demand Service”, In *Proc. of Multimedia Conference*, San Jose, CA, February 1995.
- [11] P.-F. You and J.-F. Pâris, “A Better Dynamic Broadcasting Protocol for Video on Demand”, In *Proc. of IPCCC*, pp. 84–89, Phoenix, AZ, April 2001.