Cost-optimal Dimensioning of a Large Scale Video on Demand System

David Choi
Stanford University
Stanford, CA 94305
CA 94305, USA
david.choi@stanford.edu

Ernst W. Biersack
Institut Eurecom
2229, route des Crêtes
06904 Sophia-Antipolis, France
erbi@eurecom.fr

Guillaume Urvoy-Keller
Institut Eurecom
2229, route des Crêtes
06904 Sophia-Antipolis, France
urvoy@eurecom.fr

ABSTRACT

We introduce a video distribution architecture where the prefix of the video is delivered on demand by a prefix server while the body of the video is broken in equal length segments and each segment is periodically transmitted by a central server. The combination of open-loop distribution for the video body and closed-loop multicast transmission for the video prefix makes the overall system highly scalable and very cost-efficient.

Given that video distribution architecture, we develop a detailed analytical model for the cost of delivering a video as a function of the popularity of that video. In difference to previous studies, our model includes the cost for server storage and server I/O and represents the video distribution network as an m-ary multicast tree, which allows us to precisely capture the cost for multicast transmission.

Our analytical model allows to compute the cost-optimal prefix length as well as the optimal location of prefix servers in the video distribution system. Our results show how the length of the prefix and the placement of the prefix servers depend on the video request rate. In particular, placing the prefix servers close to the clients, as was suggested in previous studies, is often not cost-optimal.

Keywords

Video streaming, content distribution network, multicast, cost model, optimal server placement.

1. INTRODUCTION

1.1 Background

Video streams such as MPEG-2 encoded video require several Mbps and providing a VOD service to a large number of clients poses high resource demands to the server and the network. The bandwidth-intensive nature of video requires efficient distribution techniques that typically serve multiple clients who request the same video at approximately the same time via a single video stream that is multicast. VoD systems can be classified in *open–loop systems* [10] and *closed–loop systems* [7, 13].

- Open loop VoD systems partition each video into smaller pieces called *segments* and transmit each segment on a separate channel at its assigned transmission rate. The first segment is transmitted more frequently than later segments because it is needed first in the playback. All segments are transmitted periodically and indefinitely. In open–loop systems there is no feedback from the client to the server, and transmission is completely one–way.
- Closed-loop systems, on the other hand, require the client to contact the server. Closed–loop systems generally open a new unicast/multicast stream each time a client or a group of clients issue a request for a video. To make better use of the server and network resources, client requests are often batched and served together with the same multicast stream.

Both, open and closed-loop schemes often incur a non-zero start-up delay[1] for the clients due to the fact that the video requested is typically not available for instantaneous playout: In case of open-loop schemes, the client must wait until he has received the beginning of the first segment; for closed-loop schemes, the server usually batches several requests that are then served by the same multicast transmission.

1.2 Contributions and Related Work

We propose a scalable and efficient video distribution architecture that combines open-loop and closed-loop mechanisms to assure a zero start-up delay. Each video is partitioned into a prefix and a suffix. The suffix is stored at a central server, while the prefix is stored at one or more prefix servers. A client who wants to view a video joins an already on-going open-loop multicast distribution of the suffix while immediately requesting the prefix of the video as a patch [11] that is sent either via unicast or multicast [6]. We develop

[1] We do not refer here to the *transmission* delay due to sending a request to a server or joining a multicast group, which we ignore here.

an analytical model for that video distribution architecture that allows to compute for a video with a given popularity the cost-optimal partitioning into prefix and suffix and the placement of the prefix servers in the distribution tree.

In contrast to previous studies (see for example [4, 8, 15]), we

- Model the network as a tree with outdegree m with l levels. In comparison, Guo et al. [8] consider only a two-level distribution architecture.
- Account in the model of the network transmission cost for the number of clients that are simultaneously served by the multicast distribution (either from the prefix server or the suffix server).
- Allow for the prefix servers to be placed at any level in the distribution tree, and not only at the last hop between client and network [8].
- Include in our cost model not only network transmission cost but also the *server* cost, which depends on both, the storage occupied and the number of input/output streams needed. While the network transmission cost is a major cost factor, the server cost must be included in the overall cost model, especially when we try to design a cost-optimal video distribution architecture. Otherwise, independent of the popularity of a video, the obvious/trivial architecture will be one where a large number of prefix servers are placed near the clients. While previous papers [8] have treated in their model the storage space of the prefix servers as a scarce resource, we feel that the cost model can be made more realistic by explicitly modeling the cost of the prefix servers.

Almeida et al. [2] considers the same problem, however the protocols used for the video delivery are different and the network delivery costs are not modeled in detail.

2. THE SYSTEM ENVIRONMENT

2.1 Prefix caching assisted periodic broadcast

Prefix caching assisted periodic broadcast[2] assumes that clients are serviced by a main central server and also by local prefix servers, which can be located throughout the network. A video is partitioned into two parts, the prefix and suffix, which can be of arbitrary proportion. The entirety of the prefix is always *viewed before* the suffix. The main idea of the broadcast scheme is that prefix and suffix transmission should be decoupled in order to transmit each most effectively. The reason why the prefix and suffix are transmitted differently is that the client must receive the prefix *immediately upon request* while the suffix need not be received until the prefix has been completely viewed.

Because the prefix must be immediately received, there is less flexibility in the choice of transmission scheme for the prefix. As a result, transmitting the prefix from the central server to each client may be costly. In order to reduce transmission costs, the prefix is stored locally at multiple prefix servers, which can more cheaply transmit to their local audiences. For the suffix, on the other hand, there is more leeway in the method of broadcast, since it needs not be received immediately. Since transmission should therefore be cheaper for the suffix, it is retained at the central server, avoiding the server costs incurred for replicating data across multiple servers.

Once specific transmission schemes for prefix and suffix have been chosen, the remaining design parameter is the length of the prefix (and suffix). The prefix length should be chosen so as to efficiently divide the workload between central server and prefix servers.

2.2 The distribution network

We assume that the distribution network is organized as an *overlay* network. An overlay network consists of a collection of nodes placed at strategic locations in existing network, e.g. the Internet. Overlay networks provide the necessary flexibility to realize enhanced services such as multicast [12] or content distribution [1] and are typically organized in an hierarchical manner. We assume that the topology of our distribution network is a dense m-ary tree with l levels (see figure 1). The central server is assumed to be at the root and the clients are lumped together at the m^l leaf nodes. The prefix servers may be placed at any level of the distribution network.

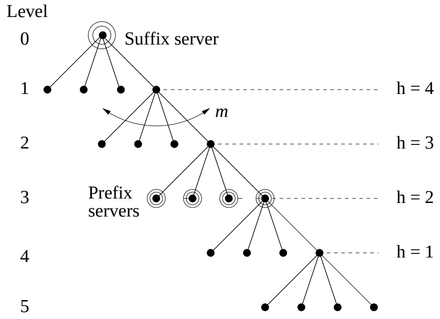


Figure 1: Video distribution network

The distribution network is assumed to support both unicast and multicast transmission. Unicast transmission occurs between a server and a single client, whereas multicast transmission occurs when multiple clients (possibly from different leaf nodes) all simultaneously receive the same single transmission from a server. We assume that for the duration of a transmission, a cost must be paid only for every link spanned between the server and its active client(s). The per-link cost may differ depending upon the specific links that are utilized.

For a multicast transmission, the cost may change over the duration of the transmission as users join and leave. Note also that if multiple clients reside at a single leaf node then the cost of multicast transmission is effectively the same as if there were only a single client at that node. For clients at different nodes, multicast still offers savings due to links shared at the higher levels by different nodes.

2.3 Prefix transmission via controlled multicast

Patching was first proposed in [11] and then extended

[2] The term broadcast is commonly used in the literature. In our model, broadcast really refers to multicast as the data are sent only over links that reach clients who are interested in receiving the video.

with the inclusion of a thresholding policy to produce **Controlled Multicast** [6]. The key idea of patching is to allow clients to share segments of a video stream when they arrive at different times. As the number of clients increases from one to several, the transmission stream is changed from a unicast stream to a multicast one so that late arrivals can still share in the remainder of the stream. In addition, however, a separate unicast stream must also be transmitted to each client after the first in order to deliver the data missed due to its later arrival.

For extremely late arrivals, the cost of the additional unicast transmission may outweigh the benefits of sharing in the remaining transmission. Controlled multicast modifies patching to allow for this scenario. Whenever a new transmission is started at time t, arriving clients are patched onto the stream until time $t + T$, where T is a **thresholding** parameter. The first client to arrive after time $t + T$ is given a brand new transmission, and all future arrivals are patched onto the new transmission instead of the old one, until the threshold time passes again and the process is repeated.

The costs of controlled multicast have been shown to increase sub-linearly with the arrival rate of requests and the length of the prefix [13]; however, the analysis assumes a network of a single link between the server and all its clients. This is the case when the prefix servers are located at the leaf nodes. Placing the prefix servers higher up in the distribution network increases the cost of transmission; however, it also consolidates the arrivals to a smaller number of servers, and thereby allows for more sharing to occur amongst clients. Furthermore, placing the prefix servers higher up in the network reduces server costs since there are fewer copies of the prefix. One contribution of this paper is an analysis of the tradeoffs in prefix server placement for controlled multicast.

2.4 Tailored periodic broadcast of the suffix

In tailored transmission [3], the suffix is broken up into segments of fixed lengths. If there are no clients then the server does not transmit. As long as there is at least one client, each segment is periodically multicast at its own transmission rate. Arriving clients receive the multicast of each segment simultaneously. Clients are not expected to arrive at the starting point of each segment; instead, they begin recording at whatever point they arrive, store the data, and reconstruct each segment as they receive the data.

3. EVALUATION

3.1 Model

We divide the costs of a VoD network into network and server costs. The network costs are proportional to the amount of network bandwidth that is used over each link between a server and its clients. The server cost is dependent upon the necessary storage, and upon the total number of input/output streams that the server(s) must simultaneously support over the network.

We will examine the expected costs over time as a function of the arrival rate λ, the prefix length (and allowable suffix delay) D, and the topology of the network. In actuality, the maximum output capability should be fixed for each server; however to facilitate analysis we will assume that any number of streams can be allocated with the costs paid on a per-stream basis.

In the following, we briefly explain our cost model [5]. The total cost of the system is given by

$$C^{system} = C^{prefix} + C^{suffix}$$

The prefix and suffix cost terms are given by

$$\begin{aligned} C^{prefix} &= C^{prefix}_{netw} + \gamma C^{prefix}_{server} \\ C^{suffix} &= C^{suffix}_{netw} + \gamma C^{suffix}_{server} \end{aligned}$$

To relate the network and the server cost, a normalization factor, γ is introduced that allows us to explore various scenarios for the cost of the servers as compared to the cost for the transmission bandwidth. For space reasons, we considered here only the values of $\gamma = 0$ and $\gamma = 1$. The case of $\gamma = 0$ corresponds to the case that only the cost for network transmission is taken into account and the cost for the servers is not considered at all (considered to be zero).

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

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

To be able to relate the cost for storage and I/O, we introduce the normalization factor β that is determined as follows: If our server has a storage capacity of d_{sto} [Megabyte] 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 (I/O is the bottleneck and no more requests can be served) or storage limited (storage volume is the bottleneck and no more data can be stored), the server cost is given as the *maximum* of $C_{I/O}$ and βC_{sto}.

To model a case where the cost for the "backbone" bandwidth is not the same as the cost for the bandwidth on the "last hop", we can set the cost for the last link to the clients to a value different from the cost for the other links. In this extended abstract, we can not derive the different cost terms. The complete derivations can be found in the appendix where we derive the prefix and suffix costs and list the formulas for all the cost terms (see table 1).

3.2 Results

The network has an out-degree $m = 4$ and a number of levels $l = 5$. All the results presented are for $\beta = 0.001$, which is a realistic value for the current disk technology such as the IBM Ultrastar 72ZX disk. The server cost is weighted by $\gamma = 1$ or $\gamma = 0$, the network per-link costs were uniform at all levels of the network, and the length of the video is $L = 90$ minutes.

The optimal values for the prefix length and cache placement in the hierarchy as a function of the request arrival rate, i.e. video popularity, are given in figures 2 and 3. For videos that are very rarely demanded ($\lambda << 1$), the prefix cache is placed at the *root* and the optimal prefix comprises the whole video of 90 minutes. Indeed, for $\lambda << 1$ the storage cost due to a replication of the prefix in multiple prefix servers is not justified and the optimal architecture is a *centralized* one. On the other hand, for videos that are popular or very popular, the optimal architecture is a *distributed* one with the server for the suffix at the root and the prefix servers closer to the clients. As the popularity λ increases, the optimal prefix length decreases since the transmission bandwidth required by the prefix server increases with the square root of the number of clients served simultaneously, while for the suffix server the transmission bandwidth required depends in the case of very high request rates only on the length of the suffix and not the number of clients served.

We plot the prefix-suffix cost division in figure 4. We see that the suffix is initially cheaper than the prefix, since the prefix length is quite long. Eventually, the suffix system becomes more cost efficient, and so usage of the prefix is reduced and the suffix costs become greater. We see that for $\lambda > 100$, the value C^{suffix} for the suffix cost only changes (increases) when the suffix length increases (cf. fig. 3(b) and fig. 4). For a given suffix length, the fact that C^{suffix} does not change with λ indicates that *all* the links of the video distribution network are active, i.e. the multicast transmission is in fact a broadcast to all leaf nodes.

Finally, we examine the case where $\gamma = 0$. Here, we completely ignore the server costs and only design for the optimal network cost. We plot the optimal prefix lengths for $\gamma = 0$ with uniform link costs in figure 5. The optimal prefix server height, which is not shown in the figure is for all request rates λ at $h = 1$, i.e. at the leaves. Placing the prefix servers at the leaves is always optimal when $\gamma = 0$, since the prefix network cost is minimized and the server costs are neglected. Since we ignore server cost, the optimal values for the prefix are much larger than for the case of $\gamma = 1.0$ (cf. fig. 3(b)). Nevertheless, as λ increases, the optimal prefix length for $\gamma = 0$ is reduced. This is due to the fact the centralized suffix system becomes so much more bandwidth efficient (despite the fact that the video must traverse 5 hops for the suffix as compared to 1 hop for the prefix) than the prefix transmission via controlled multicast.

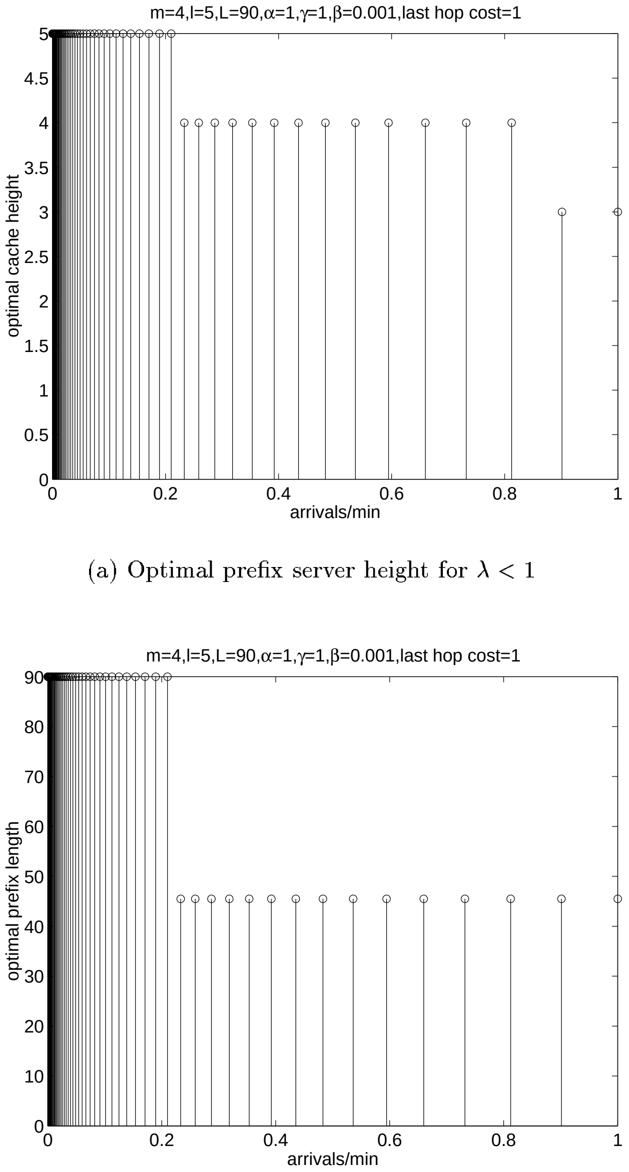
4. CONCLUSIONS AND FUTURE WORK

We have presented a video distribution architecture that combines open-loop broadcast of the suffix from a central server with a closed-loop controlled multicast patching of the prefix from a prefix server. The overall architecture is very cost-effective and highly scalable since the resource requirements for the central server are, at high request rates, independent of the number of requests and the resource requirements for the prefix servers increase only with the square root of the number of requests.

We have developed an analytical cost model for this architecture that comprises both, the network transmission and the server cost. Using this model we can determine the

- Bandwidth and streaming costs for prefix and suffix transmission
- Optimal prefix length
- Optimal position of the prefix servers.

Our model allows us to consider the cost tradeoffs related to prefix length and prefix server location as a function of video popularity. We show that, for very popular videos, it is cost-optimal to replicate the prefix in many prefix servers that are close to the clients. In this case, the prefix length is rather *small* and the suffix makes the major portion of the video. However, for videos with low request rates, it is more cost-efficient to place the prefix servers further away from the clients. In this case, the cost-optimal architecture corresponds more to a centralized patching system, with a long prefix served by only a few prefix servers that are close to the root of the distribution hierarchy.



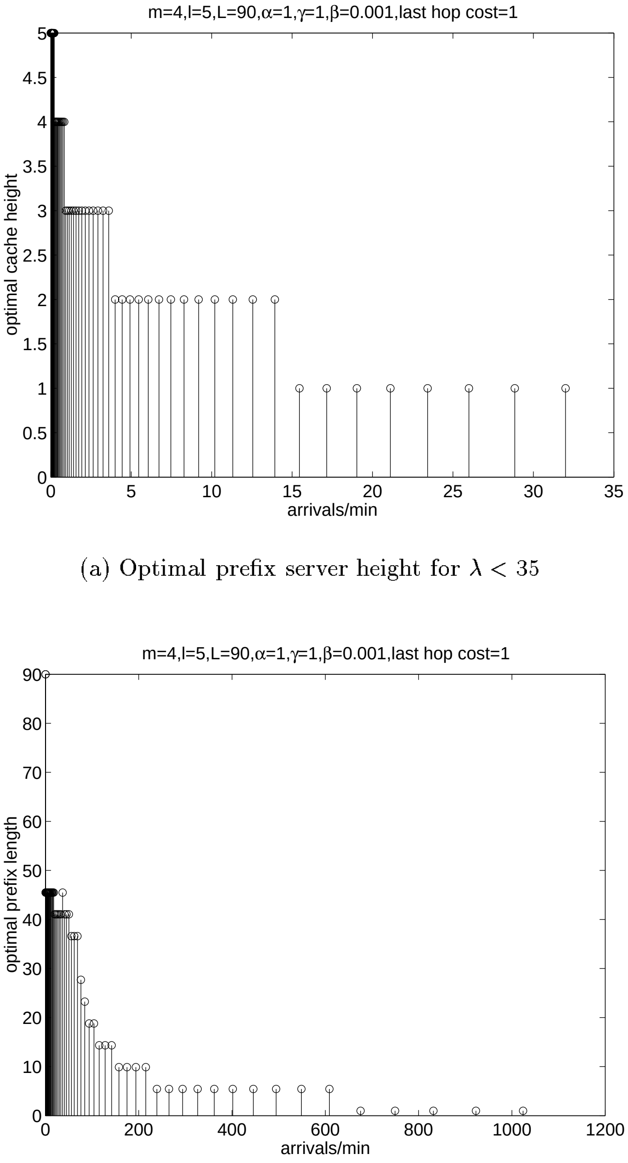
(a) Optimal prefix server height for $\lambda < 1$

(b) Optimal prefix length for $\lambda < 1$

Figure 2: Optimal prefix length and prefix server height for $\gamma = 1$, $\lambda < 1$.

These results fit well with intuition and suggest that the cost model introduced in this paper is adequate for a wide variety of scenarios. We have already started to explore some of these scenarios, such as

- Provisioning.
  The analytical model presented can be used to solve the provisioning problem for a given set of videos whose request rates are known: We just need to execute the



(a) Optimal prefix server height for $\lambda < 35$

(b) Optimal prefix length for $\lambda < 1100$

Figure 3: Optimal prefix length and prefix server height for different arrival rates, $\gamma = 1$.

model for each video separately to determine the optimal prefix length and the placement of the prefix servers.

- Video assignment problem for an existing configuration.
  Very often, the situation will be such that the video distribution system has been already deployed, i.e. the central server and the prefix server have been installed and changing the location (height) of a prefix servers is not possible. When the placement of the servers is fixed a priori, then for a given set of videos and request rates, one can compute the cost-optimal prefix length and prefix server via dynamic programming.

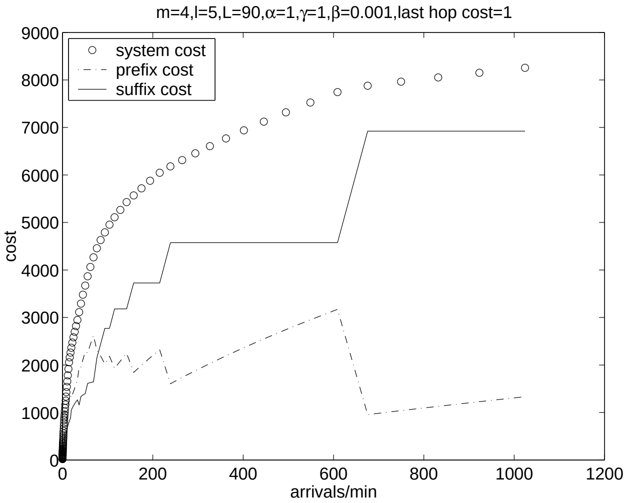


Figure 4: Breakdown of total system costs for different request rates, $\gamma = 1$.

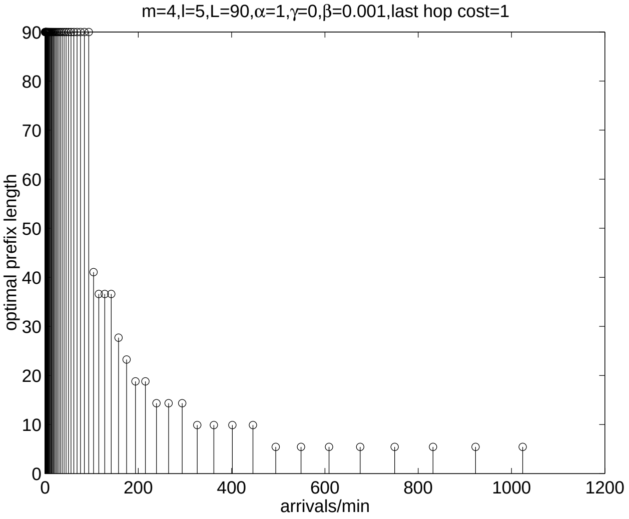


Figure 5: Optimal prefix length for different request rates, with $\gamma = 0$ and homogeneous per-link network costs.

- Evaluation of architectural choices.
  Today, digital VCRs with several tens of Gigabyte of local storage are commercially available [14]. Given local storage, one can proactively download the prefixes of the most popular vidoes directly into the VCR. We used the analytical model presented in this paper to evaluate the overall cost reduction due to the use of local storage in the VCR [9].

As further work, we will consider more general distribution trees where not all receivers are at the same distance from the root and the case of heterogeneous request patterns, where the popularity of a particular video varies among different client sub-groups.

5. REFERENCES

[1] "FreeFlow: How it Works. Akamai, Cambridge, MA, USA. Nov 1999".

[2] J. Almeida et al., "Provisioning Content Distribution Networks for Streaming Media", In *Proc.INFOCOM 2002*, June 2002.

[3] Y. Birk and R. Mondri, "Tailored Transmissions for efficient Near-Video-on-Demand Service", In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp. 226–231, June 1999.

[4] G. Chan and F. Tobagi, "Distributed Servers Architectures for Networks Video Services", *IEEE/ACM Transactions on Networking*, 9(2):125–136, April 2001.

[5] D. Choi, E. W. Biersack, and G. Urvoy-Keller, "Evaluation of Cost-optimal Architectures for VOD Servers", , Institut Eurecom, August 2001.

[6] L. Gao and D. Towsley, "Threshold-Based Multicast for Continuous Media Delivery", *IEEE Transactions on Multimedia*, 3(4):405–414, December 2001.

[7] L. Gao and D. Towsley, "Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast", In *Proceedings of IEEE Multimedia Computing Systems*, pp. 117–121, June 1999.

[8] Y. Guo et al., "Prefix Caching assisted Periodic Broadcast: Framework and Techniques for Streaming Popular Videos", In *Submitted for review*, May 2001.

[9] A. A. Hamra, E. W. Biersack, M. Jarraya, and G. Urvoy-Keller, "An Hybrid Model for a Cost-optimal Video on Demand Architecture", Technical Report, Institut Eurecom, March 2002.

[10] A. Hu, "Video-on-Demand broadcasting protocols: A comprehensive study", In *Proceedings of IEEE INFOCOM*, volume 1, pp. 508–517, Anchorage, Alaska, USA, April 2001.

[11] 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.

[12] J. Jannotti et al., "Overcast: Reliable Multicasting with an Overlay Network", In *Proc. 4-th Symp. on Operating Systems Design and Implementation*, Usenix, October 2000.

[13] S. Ramesh, I. Rhee, and K. Guo, "Multicast with Cache (Mcache): An adaptive Zero-Delay Video-on-Demand service", In *Proceedings of IEEE INFOCOM*, April 2001.

[14] TiVo, "What is TiVo: Technical Aspects", 2001.

[15] B. Wang et al., "Proxy-based Distribution of Streaming Video over Unicast/Multicast Conncetions", In *Proc. INFOCOM 2002*, June 2002.

APPENDIX

A. COSTS FOR PREFIX TRANSMISSION WITH A MULTICAST TREE

We divide the multicast tree into levels $1, \ldots, l$, where level 1 consists of the m links from the root and level l consists of the m^l links connected to the leaf nodes. Arrivals to a link at level j can be modeled as a Poisson process with parameter λ/m^j.

We first present the costs of patching with m^l patch servers at the leaves and then derive the cost of patching with a single server at the root. We next generalize the root case to the general case where the servers are placed at some level between the root and the leaves. The costs fall into three categories: network, storage capacity, and I/O capacity.

The prefix is of length D. For the root server, bandwidth costs must be paid for every stream, on every link.

We neglect the effects of network latency, even though they can be important from an operational point of view. One might treat latency effects by constraining the maximum number of hops allowed between proxy servers and their clients.

A.1 Patching at the leaves

Consider m^l different patching servers. As given in [7, 8], the average network bandwidth per unit time, under the optimal threshold, for a single server with parameter λ/m^l is $\sqrt{2D\lambda/m^l+1}-1$. As a result, the average network bandwidth C^{leaves}_{netw} for all m^l servers is simply:

$$C^{leaves}_{netw} = m^l\sqrt{2D\lambda/m^l+1} - m^l = m^{l/2}\sqrt{2D\lambda+m^l} - m^l.$$

The storage cost for m^l patch servers is $C^{leaves}_{sto} = m^l D$. The average number of I/O streams which a patch server must support is equal to the average bandwidth of the server. Therefore the I/O stream cost for m^l patch servers is $C^{leaves}_{I/O} = m^{l/2}\sqrt{2D\lambda+m^l} - m^l$

A.2 Patching at the root

Now consider a single patching server at the root of the multicast tree. We will afterwards generalize our results to the case where the server is at an arbitrary height in the tree.

A.2.1 *Bandwidth costs*

We first consider the network bandwidth costs of a single root server. A single server at the root combines many small arrival streams (to the leaves) into a single large arrival stream (to the root); this should lower costs, since patching is sublinear in the arrival rate. Patching achieves sublinear cost performance because it promotes efficiency through shared streams; by combing all the arrivals into a central server we increase the possibility for sharing. However, this is counterbalanced by the fact that sharing is no longer free; if two clients share the same I/O stream but reside on different leaves, a separate network cost must be paid of each of them. Of course, if clients are already active at every leaf node, then no new network costs must be paid for any future arrivals. However, this scenario is unlikely even for high arrival rates, because high arrival rates produce short threshold times, in order to reduce the length of the unicast streams.

Let t_i be the time of the ith complete multicast transmission of the prefix, without any patching. Arrivals between times t_i and t_{i+1} will share from the multicast transmission at time t_i and will each receive a separate unicast transmission for the data which was missed. We can divide the patching process up into separate renewal cycles $(t_1 t_2], (t_2 t_3], \ldots$ which are independent and identically distributed in their usage of bandwidth. We analyze the bandwidth usage over a single renewal process.

Given the threshold time T, on average there will be $T\lambda$ arrivals which will each need partial transmission of the prefix in unicast. The average length of the unicast transfer will be $T/2$ since the arrivals are uniformly distributed over time. Finally a bandwidth cost must be paid of every link on the path between client (at the leaf) and the root server. As a result, the total bandwidth expended for the unicast transmissions over one renewal cycle is

$$C_{netw}^{unicast} = \frac{lT^2\lambda}{2}.$$

Each arrival will also share from a single multicast network stream. A price must be paid for every link in use. Divide the multicast tree into levels $1, \ldots, l$, where level 1 consists of the m links from the root and the links at level l are the ones which connect to the m^l leaf nodes. Given a link in level j, let τ_j be the duration of time in which the link is active. For the multicast stream, a link is active from the time of the first arrival (before time T to that link to the end of the prefix at time D). Arrivals to a link at level j can be modeled as a Poisson process with parameter λ/m^j

As each renewal cycle begins with the activation of a stream between the root and a single client, we know that one link at each level will be active at time zero. Therefore $\tau_j = D$ with probability $1/m^j$. We will now write out an expression for $\mathbb{E}[\tau_j]$:

$$\begin{aligned}\mathbb{E}[\tau_j] &= D\mathbb{P}\{\tau_j = D\} + \mathbb{E}[\tau_j | \tau_j \neq D]\mathbb{P}\{\tau_j \neq D\} \\ &= D\frac{1}{m^j} + \mathbb{E}[\tau_j | \tau_j \neq D]\frac{m^j - 1}{m^j}.\end{aligned}$$

Given a Poisson process with parameter λ, the time of first arrival will have an exponential distribution $\lambda e^{-\lambda t}$, and a cumulative distribution $F(t) = 1 - e^{-\lambda t}$. We evaluate $\mathbb{E}[\tau_j | \tau_j \neq D]$,making use of the fact that $\int_0^T t\lambda e^{-\lambda t} dt = \int_0^T e^{-\lambda t} - e^{-\lambda T} dt$.

$$\begin{aligned}\mathbb{E}[\tau_j | \tau_j \neq D] &= \int_0^T (D - t)\lambda e^{-\lambda t} dt \\ &= \int_0^T D\lambda e^{-\lambda t} dt - \int_0^T t\lambda e^{-\lambda t} dt \\ &= D(1 - e^{-\lambda T}) - \int_0^T F(T) - F(t) dt \\ &= D(1 - e^{-\lambda T}) - \int_0^T F(T) - F(t) dt \\ &= D(1 - e^{-\lambda T}) - \int_0^T e^{-\frac{\lambda}{m^j}t} - e^{-\frac{\lambda}{m^j}T} dt \\ &= D(1 - e^{-\lambda T}) - (-\frac{m^j}{\lambda}e^{-\frac{\lambda}{m^j}t} - e^{-\frac{\lambda}{m^j}T} t)\Big|_{t=0}^{T} \\ &= D(1 - e^{-\lambda T}) - \frac{m^j}{\lambda} + \frac{m^j}{\lambda}e^{-\frac{\lambda}{m^j}T} + Te^{-\frac{\lambda}{m^j}T}.\end{aligned}$$

Plugging in to $\mathbb{E}[\tau_j]$ produces

$$\begin{aligned}\mathbb{E}[\tau_j] &= D\frac{1}{m^j} + \mathbb{E}[\tau_j | \tau_j \neq D]\frac{m^j - 1}{m^j} \\ &= D(1 - e^{-\lambda T}(1 - \frac{1}{m^j})) - (1 - e^{-\frac{\lambda}{m^j}T})\frac{m^j - 1}{\lambda} \\ &\quad + Te^{-\frac{\lambda}{m^j}T}\frac{m^j - 1}{m^j}.\end{aligned}$$

By summing over all the links in the tree we find the total multicast cost $C_{netw}^{multicast}$:

$$C_{netw}^{multicast} = \sum_{j=1}^{l} m^j \mathbb{E}[\tau_j],$$

and the average network bandwidth cost C_{netw}^{root} can be found by dividing by the average duration of each renewal cycle $T + 1/\lambda$.

$$\begin{aligned}C_{netw}^{root} &= \frac{C_{netw}^{multicast} + C_{netw}^{unicast}}{T + 1/\lambda} \\ &= \frac{\sum_{j=1}^{l} m^j \mathbb{E}[\tau_j] + lT^2\lambda/2}{T + 1/\lambda}.\end{aligned}$$

A.2.2 *Server costs for root patching*

It is easy to see that the storage cost C_{sto}^{root} of a single root server will be D. The I/O stream cost must be paid for the output capabilities of each server, i.e. the number of input/output streams which a server can simultaneously maintain. The average number of streams for a root server is equivalent to (see [6])

$$C_{I/O}^{root} = \lambda\frac{2D + \lambda T^2}{2 + 2\lambda T}$$

A.3 Varying the height of the patch server

We now generalize the costs to the case where the proxy servers are placed at any level in the network tree. By placing the patch servers at some level h in the tree where $0 > h > l$, we divide the arrival process between m^{l-h} servers, each of which can be the considered the root of a network tree with height h. We need only therefore consider the root server case for a tree of the proper height h, with arrival rate λ/m^{l-h}, and then multiply the costs by the number of servers m^{l-h}. The resulting formulas are listed in table 1.

B. COSTS FOR SUFFIX TRANSMISSION WITH A MULTICAST TREE

The costs once again fall into three categories: bandwidth, storage capacity, and streaming capacity.

It has been shown in [3] that the total transmission rate for all the segments can be minimized to $ln(1 + \frac{L-D}{D}) = ln(\frac{L}{D})$, where D is the maximum delay allowed for the initial segment (equal to the prefix length) and $L - D$ is the total length of the suffix. The bandwidth cost is equal to the transmission rate $ln(L/D)$ multiplied by the average number of active links, which we will now calculate. For the periodic broadcast, each arrival is serviced for the same amount of time $L - D$. We assume that all segments are multiplexed to a single multicast channel. As a consequence, each client will consume a bandwidth of $ln(L/D)$ during all the transmission of the suffix. If one multicast channel is dedicated to each segment, the bandwith consumption could be reduced; the client being connected only to channels corresponding to segments not yet viewed. However, this reduction in bandwidth cost comes at the expense of a more complex multicast transmission and a complex synchronisation between channels. This study is left for future work. From queuing theory, it can be shown that given an expected service time $\mathbb{E}[T_s]$ and memoryless arrivals with parameter λ, the probability of n jobs simultaneously in progress is given by

$$\mathbb{P}\{n \text{ jobs}\} = \frac{e^{-\lambda \mathbb{E}[T_s]}(\lambda \mathbb{E}[T_s])^n}{n!},$$

which is a Poisson distribution. This result can be found through the derivation of the Erlang call-blocking formula commonly used in telecommunications. Arrivals to a link at level j are memoryless with parameter λ/m^j. Define $P(j)$ as the probability that a link at level j has any requests.

$$P(j) = 1 - e^{-\frac{\lambda(L-D)}{m^j}}.$$

The expected number of active links at any given time is therefore

$$\mathbb{E}[\text{active links}] = \sum_{j=1}^{l} m^j P(j),$$

And the bandwidth is

$$C_{netw}^{suffix} = ln(\frac{L}{D}) \sum_{j=1}^{l} m^j P(j)$$

Because the suffix is continuously and periodically broadcast and does not change with the arrival rate, as long as there is at least one user (if there are no users, the central server will not send the video). However, the number of output channels does vary with the length and delay. The I/O stream cost is equal to the rate $ln(\frac{L}{D})P(0)$, where $P(0)$ is the probability that there is at least one user active at the root. The storage cost is proportional to the length of the suffix, which is $L - D$.

Cost terms	
C_{netw}^{prefix}	$m^{l-h} \frac{\frac{hT^2\lambda}{2m^{l-h}} + \sum_{j=1}^{h} m^j \mathbb{E}[\tau_j]}{T + m^{l-h}/\lambda}$ $(\mathbb{E}[\tau_j] = D(1 - e^{-\frac{\lambda}{m^j m^{l-h}}T}(1 - \frac{1}{m^j}))$ $-(1 - e^{-\frac{\lambda}{m^j m^{l-h}}T})\frac{m^{l-h}(m^j - 1)}{\lambda}$ $+Te^{-\frac{\lambda}{m^j m^{l-h}}T}\frac{m^j - 1}{m^j})$
C_{sto}^{prefix}	$m^{l-h} D$
$C_{I/O}^{prefix}$	$\lambda \frac{2D + \lambda T^2/m^{l-h}}{2 + 2\lambda T/m^{l-h}}$
C_{netw}^{suffix}	$ln(\frac{L}{D}) \sum_{j=1}^{l} m^j (1 - e^{-\frac{\lambda(L-D)}{m^j}})$
C_{sto}^{suffix}	$L - D$
$C_{I/O}^{suffix}$	$ln(\frac{L}{D})(1 - e^{-\lambda(L-D)})$

Table 1: Summary of cost terms (l levels, prefix servers at height h)