Prefix Caching assisted Periodic Broadcast for Streaming Popular Videos[1]

Yang Guo, Subhabrata Sen†, and Don Towsley
Computer Science Department
University of Massachusetts
Amherst MA 01003
†AT&T Labs-Research
Florham Park, NJ 07932

Abstract— **The bandwidth-intensive and long-lived nature of high quality digital video makes it a challenging problem to transmit such video over the Internet. In this paper, we propose a scalable and flexible framework integrating proxy-based prefix caching with periodic broadcast of the suffix of a video from the server, for efficiently streaming a set of popular videos to a large number of asynchronous clients. We develop a methodology for (i) determining appropriate prefix and suffix transmission schemes based on a principle of decoupling the two transmissions from each other, and (ii) optimally allocating the proxy buffer space among the set of videos. A buffer allocation algorithm is presented that minimizes the aggregate bandwidth usage on the server-proxy path. Our studies show that our approach yields a buffer allocation close to the optimal solution minimizing both server-proxy and proxy-client path bandwidth usage for practical settings where the proxy-client path bandwidth is much cheaper than the long-haul server-proxy path bandwidth. When the proxy buffer is allocated to a set of videos using our scheme, a total buffer space of just $5-20\%$ of the video repository is adequate to realize substantial reductions in the aggregate bandwidth usage on the server-proxy path.**

I. INTRODUCTION

The bandwidth-intensive nature and long-lived characteristics of digital video make transmission bandwidth a major limiting factor in the widespread streaming of such content over the Internet. For popular clips, the client population is likely to be large, with different clients asynchronously issuing requests to receive their chosen media streams. Different video clips can have very different sizes (playback durations) and popularities. A challenging problem is developing techniques for bandwidth-efficient distribution of heterogeneous videos to such a large, asynchronous client population.

Transmission schemes such as periodic broadcast and patching [1–5] use multicast or broadcast connections to reduce bandwidth usage, while providing a guaranteed bound on a client's playback startup latency. An orthogonal technique for reducing server loads, network traffic and access latencies is the use of proxy caches [6, 7]. In particular, caching an initial prefix of the video [7] has a number of advantages including shielding clients from delays and jitter on the server-proxy path, while reducing traffic along that path.

Fig. 1 depicts a Streaming Content Distribution Network (SCDN) considered in this paper. We explore integrating proxy prefix caching, periodic broadcast and patching for bandwidth-efficient delivery of popular videos to multiple asynchronous clients over such a SCDN. In addition to providing low startup delays and efficient bandwidth use, it is desirable that such a distribution scheme is sufficiently flexible and robust to accommodate the heterogeneities and changing dynamics inherent in an Internet environment. For example, the request rate for a particular video may vary with time, and the relative popularities of the videos may vary across different proxies. A flexible distribution scheme would be able to to cater to the particular needs of different local client populations, while still making efficient use of the server-proxy and proxy-client network bandwidths.

Key challenges in realizing such a distribution scheme include determining appropriate transmission schemes for the suffix (served from the server), the prefix (served from the proxy), as well as proxy buffer allocation.

- We develop a two-step approach for resolving these issues. First, we determine the appropriate suffix transmission scheme (periodic broadcast) and corresponding proxy buffer allocation that minimizes the aggregate bandwidth usage on the server-proxy path. Then, for each proxy, an appropriate prefix transmission scheme (combination of patching and periodic broadcast) is determined, based on the relative popularity of a video among clients of that proxy.
- We present a greedy algorithm to determine the allocation of the proxy buffer space that the aggregate bandwidth usage on the server-proxy network path is minimized.

Decoupling the suffix and prefix transmissions enables different proxies to use different prefix transmission schemes, while sharing the same common suffix transmission from the server. This also allows a proxy to dynamically adapt its prefix transmission scheme to small variations in the local demand for any video without requiring changes to the global suffix transmission scheme. Our results show that for practical settings where bandwidth on the local proxy-client path is significantly cheaper than the bandwidth on the long-haul

---

[1] This research was supported in part by the National Science Foundation under NSF grants EIA-0080119, NSF ANI-9973092, and ANI9977635. Any opinions, findings, and conclusions or recomendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

server-proxy path, our approach results in buffer allocations and end-end bandwidth costs which are close to what would be achieved under an optimal approach that attempts to minimize the server-proxy and proxy-client bandwidth costs together.

Our studies show that for a single video, our approach of *making the first segment of the suffix broadcast equal in size to the prefix* can substantially reduce bandwidth usage on the long-haul path. When the proxy buffer is allocated to a set of videos using our allocation scheme, a total buffer space of just $5-20\%$ of the video repository is adequate to realize substantial reductions in the aggregate bandwidth usage on the server-proxy path. Furthermore, the choice of a particular periodic broadcast scheme does not significantly impact the number of server channels required provided that the proxy can cache at least $10-20\%$ of the video repository.

This paper complements other recent works that combine caching with scalable transmission of continuous media [8]. Evaluations (Section IV-D) show that with the same available proxy buffer space, our scheme results in a much lower bandwidth usage on the server-proxy path than the scheme proposed in [8].

The remainder of the paper is organized as follows. Section II introduces key concepts and terminology. Sections III present our composite delivery scheme and proxy buffer allocation algorithm. Section IV evaluates the scheme. Finally, Section V concludes the paper and describes ongoing work.

II. PROBLEM SETTING

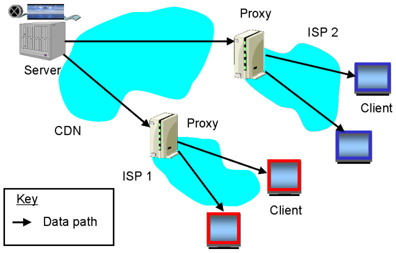


Fig. 1. An Internet Content Distribution Network

Consider a single proxy and the client population it serves. We assume that both the server-proxy network paths and proxy-client network paths are multicast capable, and that clients always request playback from the beginning of a video. The prefix is streamed to the client using a *prefix transmission* scheme. The remainder of the video (suffix) is delivered using a corresponding *suffix transmission* scheme. The prefix and suffix transmission schemes together constitute the transmission scheme for a video for the clients of a particular proxy.

We next provide a formal model of the system, and introduce notation and key concepts that will be used in the remainder of the paper. We consider a multimedia server with a repository that includes K popular videos. We assume Constant-Bit-Rate (CBR) videos with identical playback rates. Video i is l_i seconds long, and the first $x_i l_i$ seconds of it are cached at the proxy, $0 < x_i \leq 1$. Henceforth x_i will be referred to as the prefix size of video i. Note that while an origin server is typically an expensive high-end server or server farm with substantial storage (terabytes), processing and IO bandwidth capacity, the far more numerous edge proxies are more likely to be relatively inexpensive devices with more limited capabilities. In addition, a single proxy is likely to serve multiple origin servers, which will contend for the proxy resources. In this paper, we assume that B seconds worth of proxy storage is available to the server. The videos cached at the proxy cannot exceed this storage constraint, that is, $\sum_{i=1}^{K} x_i l_i \leq B$.

An important goal in our scheme is to minimize the transmission bandwidth requirement on the server-proxy path, i.e., the metric $C = \sum_{i=1}^{K} c_i$, where c_i is the number of server channels for video i. Once the prefix allocation is known, the individual video prefixes are stored at the proxy, in advance of client requests, based on the proxy buffer allocation.

For ease of exposition, we assume zero propagation delay in the rest of the paper. However, our results are easily extended to account for a bounded propagation delay. We also assume the client has sufficient buffer space and network bandwidth to accommodate an entire video clip.

We use periodic broadcast for suffix delivery [1–3]. In this paper we focus on broadcast schemes that divide a video into segments of increasing length that are transmitted over channels of the same bandwidth. However, our approach is applicable to other types of broadcast schemes. We associate with each segment an integer $f(n)$, s.t. the length of the i-th segment is $f(i)l/\sum_j f(j)$, and $f(1)$ is taken to be one. We shall consider some representative schemes which use the following segment size progressions: Skyscraper: [1,2,2,5,5,12,12, $\cdots$], Dynamic skyscraper: [1,2,2,4,4,8,8, $\cdots$], GDB(3): [1,2,4,6,8,12,16,$\cdots$], GDB(4): [1,2,4,8,14,24,40,$\cdots$], GDB(5): [1,2,4,8,16,30,56, $\cdots$], GDB(6): [1,2,4,8,16,32,62,$\cdots$].

III. PREFIX CACHING ASSISTED PERIODIC BROADCAST

In this section we describe our video distribution framework, and present some key intuitions and design principles guiding our approach. We propose the following two-step approach to determining the suffix and prefix transmission schemes as well the proxy prefix buffer allocation to each video. **Step 1.** For each video, we determine the appropriate suffix transmission scheme and corresponding proxy buffer allocation such that the aggregate bandwidth usage on the server-proxy path is minimized. **Step 2.** for each proxy, an appropriate prefix transmission scheme (combination of patching and periodic broadcast) that makes efficient use of bandwidth on the proxy-client path, is determined, based on the relative popularity of a video among clients of that proxy.

A key feature of this approach is the decoupling of the local (prefix) transmission from the long-haul path (suffix) transmission. The rationale behind this is as follows:

- Typically it is more expensive to transmit a unit of data from the remote server to the proxy, than from the proxy to the client. In section IV-C, we shall show that the total bandwidth used to deliver a single video is close to the long haul bandwidth requirement, if the cost of transmitting a unit of data locally is relatively small (compared to the long haul cost). Thus our decoupling approach yields a solution that is close to the globally optimal solution in many practical settings, while still enjoying the benefits of simplicity, flexibility, and robustness against the presence of incomplete request rate information.
- The local transmission cost is closely related to the local client request rate, which is not easy to obtain and difficult to predict, making the accurate estimation of local delivery cost practically infeasible. Furthermore, there can be tremendous heterogeneity among proxies. It may be better to let a proxy take the responsibility to determine how to deliver the prefix locally.

The scalability of our technique derives from the combination of periodic broadcast with proxy prefix caching, as well as from the minimization of the long-haul server-proxy path bandwidth. The flexibility comes from decoupling the prefix and suffix transmission from each other, We next present key components of our distribution scheme, first describing our technique for determining the prefix and suffix transmission schemes for a given video, and then addressing the multiple video proxy buffer allocation problem.

A. Distributing a single video

In our framework, the proxy and the server are responsible for serving a prefix and corresponding suffix respectively, of the video. The server receives requests for the suffix from a large number of clients across multiple proxies. We therefore elect periodic broadcast as the suffix transmission technique. For a given prefix, increasing the first suffix segment size can reduce the number of suffix segments and hence the total suffix transmission bandwidth. Note that in the absence of prefix caching, the length of the first segment would determine the worst case startup delay for any client. In our scheme, as a separate prefix transmission scheme is used for the initial part of the video, a large first segment does not incur a large startup delay. To ensure seamless transition when the client switches from playing back the prefix to the suffix, it should start receiving the suffix before the end of prefix is reached. To guarantee this, the first segment of the suffix cannot be larger than the prefix. We therefore select the first segment to equal the prefix length. The combination of these effects is illustrated in Fig. 2, where skyscraper broadcast is used to deliver the suffix, and the first suffix segment is equal in length to the prefix. If the proxy can cache 20% of the video, 3 server channels are needed. If the prefix reaches 30%, only two server channels are required.

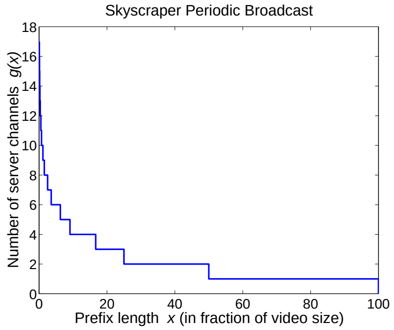


Fig. 2. Number of server channels vs. prefix size

The next question is to design appropriate bandwidth-efficient prefix transmission schemes that can support playback with low delays. Several existing schemes are potential candidates, including patching, merging and selective catching [9]. The last scheme chooses either patching or a combination of patching and periodic broadcast, based on which scheme is more bandwidth-efficient for a given local request rate. In our evaluations, we shall use selective catching as the candidate prefix transmission scheme.

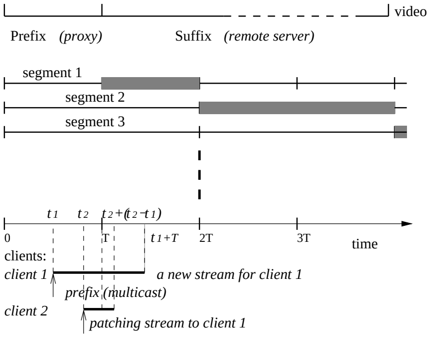


Fig. 3. Prefix-caching assisted periodic broadcast

Fig. 3 illustrates an example scenario where the suffix is partitioned into several segments (we show the first three segments) which are delivered using periodic broadcast, and the proxy uses patching to deliver the prefix. Client 1 arrives at time t_1, the proxy initiates a new stream to transmit the prefix to it. At time T, client 1 begins receiving the suffix starting from segment 1 (shaded segments). Client 2 for the same video arrives at time t_2 while the proxy is still multicasting the prefix, and taps into this ongoing transmission. Simultaneously the proxy sends a patch, $[0, \quad t_2 - t_1]$, to client 2. Notice that client 2 will also begin to receive the suffix at time T and receive the shaded segments at the same time as client 1. There is a period of time during which the client simultaneously receives both the prefix and the suffix. Thus the number of channels the client needs to listen to simultaneously, in the worst case, is the sum of the worst case number of channels required to obtain the prefix and suffix. In a longer version of the paper [10], we develop appropriate prefix and suffix transmission schemes where the client needs to listen to at most 2 channels simultaneously.

B. *Optimal proxy buffer allocation*

The next question is *how to allocate a limited proxy buffer among multiple videos so as to minimize aggregate transmission costs*. We now present a greedy buffer allocation algorithm that minimizes the long-haul path transmission bandwidth. The more detailed illustration of the algorithm and the optimality proof is included in [10].

In the first step, the proxy buffer is evenly distributed to each video. This gives us a feasible initial point. In the second step, we minimize the buffer usage without changing the total number of long-haul channels required. More specifically, we identify video i (video j) that has the minimum increase (maximum decrease) of prefix size to multicast video using one less (more) channel. If the increase of video i is less than the decrease of video j, we offer more buffer to video i so that it used one less channel while taking away some buffer from video j so that it uses one more channel. We call this "channel swapping". The result of channel swapping is that some proxy buffer is saved without changing the total number of channels required. Channel swapping is continued until no further savement on proxy buffer can be achieved. Finally, the saved buffer space from step 2 is allocated. At each round, we pick the video that requires the least buffer to use one less channel and give it the buffer. This process ends until the proxy buffer is used up.

IV. EVALUATION

We first evaluate the impact of available proxy buffer size on the performance of our scheme. We next investigate how the choice of periodic broadcast scheme impacts bandwidth savings. We further show that the buffer allocation that minimizes the long-haul bandwidth is close to the optimal solution that minimizes the sum of long-haul bandwidth and local delivery bandwidth usage. Finally, we compare our distribution technique to the scheme proposed in [8].

We consider a set of twenty 100 min. long videos, each with a playback rate of 2 Mbps. We expect similar conclusions to hold for different numbers of videos.

A. *Proxy Buffer Size*(B)

Fig. 4 depicts the long-haul path bandwidth requirement as a function of the proxy buffer size. The bandwidth requirement decreases, initially rapidly and then more gradually, with increasing proxy buffer size. The behavior suggests that most of the bandwidth saving can be achieved in the buffer region about $6-23\%$ of the total video repository.

Fig. 5 depicts the number of videos that can be supported as a function of the buffer size, given a fixed number of available long-haul path channels (each with a bandwidth equal to the playback rate). For instance, a proxy buffer that can hold 2.5 videos can support 20 videos with 80 remote server channels, and a proxy buffer of 6 videos is needed to support 30 videos with 80 remote server channels.

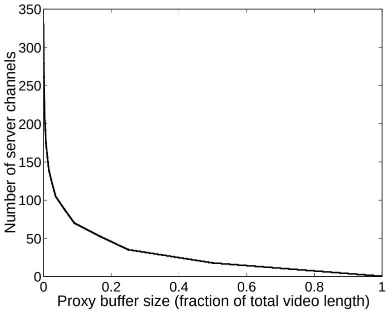


Fig. 4. Bandwidth vs. buffer size (a set of 20 videos)

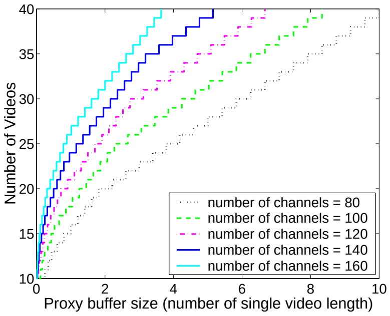


Fig. 5. Number of videos vs. buffer size

Since the videos are all of the same length, the buffer is roughly evenly distributed among them. We further investigate the proxy buffer allocation with a set of videos of different sizes. We consider a set of 20 videos with lengths 20 min, 30 min, ..., 210 min. Fig. 6 shows how the proxy buffer is allocated among the different videos. The videos are numbered in order of increasing length. We represent the prefix size as a fraction of the video length. One observation is that a larger fraction of shorter videos are stored at the proxy. When we increase the proxy size from 5%, to 30% of the total video repository, this trend becomes more significant. This phenomenon is understandable since allocating a given amount of buffer to a shorter video has a greater chance to reduce more server channels than to a longer video.

We also compare the optimal buffer allocation scheme with a naive buffer allocation scheme, where the buffer is evenly divided among the videos. The optimal buffer allocation always outperform the naive scheme; and the difference becomes larger as the prefix increases in size. In the region where the proxy buffer is about $10-20\%$ of the video size, the optimal allocation scheme reduces the number of required remote server channels by about 18% over the naive scheme.

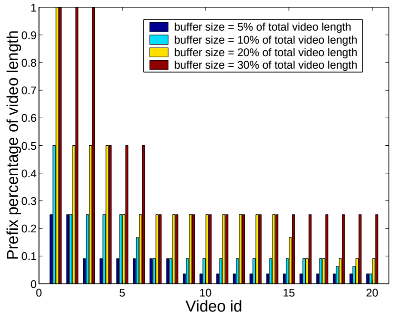


Fig. 6. Prefix percentage of video length (a set of 20 video with different length)

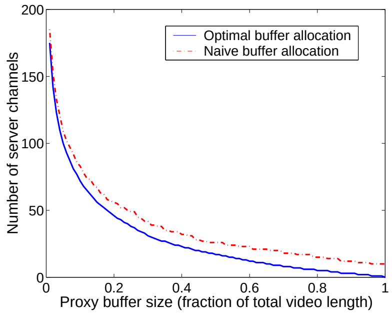


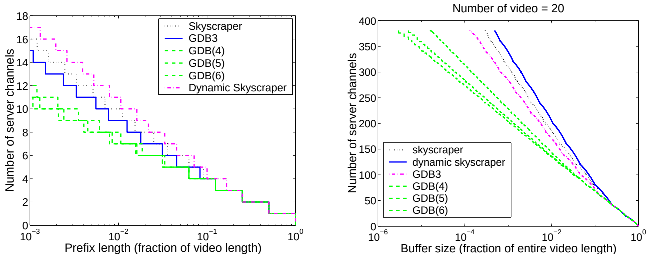
Fig. 7. Optimal buffer allocation vs. even buffer allocation

B. *Choice of periodic broadcast scheme*

Fig. 8 depicts the number of server channels vs. the prefix size for several periodic broadcasting schemes, such as skyscraper, GDB3, GDB(4), GDB(5), GDB(6), and dynamic skyscraper. Note that if the prefix size is around 20% of the video size, then all schemes need the same number of server channels, more specifically, three server channels. The reason for this is that the first three segment sizes are very similar across all the the periodic broadcast schemes, and when the prefix size is around 20%, under our approach of having the first segment of the suffix equal to the prefix, the suffix has three segments. The graphs show that a proxy buffer size of around $10 - 20\%$ of the video repository, the difference between the different schemes is small. Only for much small buffer sizes, more aggressive broadcast schemes tend to perform better. We thus recommend choosing the most aggressive scheme that accommodates the client's buffer and bandwidth constraints.

C. *Global optimization vs. decoupling approach*

In Section III, we proposed a hierarchical two-step approach for determining the optimal proxy buffer allocation, i.e., we choose $\{x_k\}$ so as to minimize $\sum_k g(x_k)$, ignoring the prefix delivery cost. Ideally we may want to perform a global optimization, that is, $\min_{\{x_k\}} \sum_k \{g(x_k) + \beta C_{local}(x_k, \lambda_k)\}$, where β is a weight placed on the local bandwidth and $C_{local}(x_k, \lambda_k)$ is the average bandwidth for both prefix and suffix transmission over the local network path, given prefix size x_k and request rate λ_k for video k. Since the proxy is typically much closer to the clients than the remote server, β is expected to be much smaller than one. Solving this global optimization problem results in the minimum global cost, but requires client request rate information for each video at every proxy. Let us see how our approach compares with the globally optimized solution in a practical setting.



(a) single video case

(b) a set of 20 videos

Fig. 8. Comparing periodic broadcast schemes

We consider the single video case. Fig. 9 depicts the average bandwidth requirement vs. prefix size for a 100 min. video for different values of β. We use the skyscraper scheme to deliver the suffix, and selective catching for prefix transmission. We choose a request rate of 60/min. We only plot the points at which the number of server channels decreases. The optimal buffer allocation must occur at these discrete points.

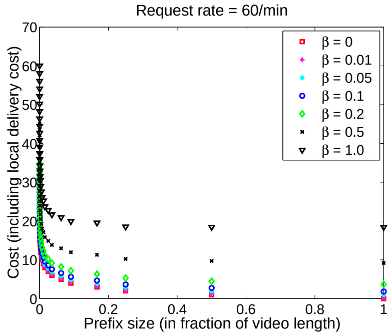


Fig. 9. Average bandwidth requirement vs. prefix size for a video of 100 mins

One observation is that the cost function is monotonically decreasing so long as $\beta < 1$, i.e., the larger the fraction of the video cached at the proxy, the less bandwidth is needed to deliver the video. This supports our intuition that we should make full use of the proxy buffer in most practical settings.

Another observation is that the total cost increases quickly once β becomes larger than 0.1, which suggests that, if possible, we should place the proxy as close to the end clients as possible.

If $\beta = 0$, only long-haul path bandwidth is considered, corresponding to the objective function used in our approach. The costs with $\beta \in [0.01,\ 0.05]$ are close to the cost for

$\beta = 0$. Hence for small β, the cost of the buffer allocation obtained by our greedy algorithm will be close to the global optimal solution, while being simple, flexible, and robust to the uncertainties about the request rate.

We also computed the costs for request rates varying from 1/min to 500/min. For the sake of the brevity, these are not presented here. We find that the smaller the request rate, the larger is the range of values of β for which the cost function is close to that for $\beta = 0$. But even for a request rate of 500/min, the cost function with β equal to 0 and 0.01 are close.

D. Comparison with optimized regional caching

Optimized regional caching (ORC) [8] uses dynamic skyscraper broadcast to initially segment the video, and categorizes some segments as leading segments. The proxy is allowed to cache the entire video, the entire leading segments, or nothing. An analytical model is then used to determine the cache allocation that minimizes the end-end delivery cost, assuming that the client request rate information at all the proxies is known beforehand.

We next compare our approach with ORC from two main perspectives: (1) proxy buffer and network bandwidth usage, and (2) the objective function used in the optimization model. Fig. 10 depicts the number of server channels required given a fixed amount of proxy buffer allocated to a 100 min long video. For fairness, we also use dynamic skyscraper broadcast for suffix delivery in our approach. For optimized caching, we choose the first segment size to be 0.1min, 1 min, and 1.5 mins respectively. Recall that in [8], the entire leading segments have to be cached in proxy. Here, for the sake of comparison, as many of the segments as possible are placed in the proxy buffer. The number of server channels is equal to the number of channels used to deliver the part that cannot be cached in the proxy.

Prefix-caching assisted periodic broadcast outperforms ORC for almost all proxy buffer sizes. For instance, when the proxy buffer is around 20% of the video size, our approach requires only $50 - 60\%$ of the long-haul path bandwidth required for ORC.

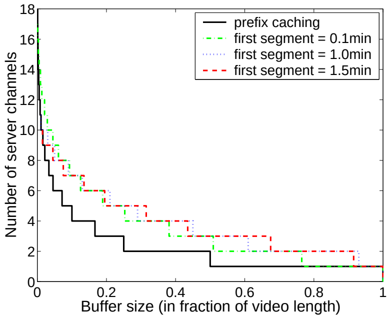


Fig. 10. Comparison of prefix caching assisted periodic broadcast vs. dynamic skyscraper scheme

We next compare the objective function used in these two works. In prefix caching assisted periodic broadcast, our goal is to minimize the number of server channels (or long-haul path bandwidth), where no client request rate information is needed. The heterogeneity of the local requests is handled by the proxy and the local prefix delivery scheme. In contrast, the ORC approach assumes prior knowledge of client request rates at the different proxies, and incorporates this information in the objective function whose goal is to minimize the aggregated end-end bandwidth cost. Our solution is close to the global optimal for many practical settings, and more robust to the uncertainty in the local information, as discussed before.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a prefix caching assisted periodic broadcast scheme and developed a methodology for (i) optimally allocating the proxy buffer space among a set of popular videos and (ii) choosing appropriate prefix and suffix transmission schemes based on the principle of decoupling the prefix and suffix transmissions from each other. We presented a greedy algorithm to determine the allocation of the proxy buffer space among the different videos. The framework is scalable, flexible, and supports popular video streaming with small playback delay. Our study also shows that the choice of periodic broadcast schemes doesn't have significant impact. We are further exploring this research space along a number of directions, such as, (1) extending our work to support VBR video delivery, and (2) exploring our framework in a realistic network setting [11].

REFERENCES

[1] A. Hu, "Video-on-demand broadcasting protocols: A comprehensive study," in *Proc. IEEE INFOCOM*, April 2001.

[2] K. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," in *Proc. ACM SIGCOMM*, September 1997.

[3] L. Gao, D. Towsley, and J. Kurose, "Efficient schemes for broadcasting popular videos," in *Proc. Inter. Workshop on Network and Operating System Support for Digital Audio and Video*, July 1998.

[4] Y. Cai, K. Hua, and K. Vu, "Optimizing patching performance," in *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, January 1999.

[5] S. Sen, L. Gao, J. Rexford, and D. Towsley, "Optimal patching schemes for efficient multimedia streaming," in *Proc. Inter. Workshop on Network and Operating System Support for Digital Audio and Video*, 1999.

[6] R. Tewari, H. M. Vin, A. Dan, and D. Sitaram, "Resource-based caching for Web servers," in *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, January 1998.

[7] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in *Proc. IEEE INFOCOM*, April 1999.

[8] D. Eager, M. Ferris, and M. Vernon, "Optimized caching in systems with heterogeneous client populations," in *Performance Evaluation, Special Issue on Internet Performance Modeling*, September 2000.

[9] L. Gao, Z. Zhang, and D. Towsley, "Catching and selective catching: Efficient latency reduction techniques for delivering continuous multimedia streams," in *Proc. ACM Multimedia*, 1999.

[10] Y. Guo, S. Sen, and D. Towsley, "Prefix caching assisted periodic broadcast: Framework and techniques to support streaming for popular videos," tech. rep., UMass CMPSCI Technical Report TR 01-22, 2001.

[11] M. K. Bradshaw, B. Wang, S. Sen, L. Gao, J. Kurose, P. Shenoy, and D. Towsley, "Periodic broadcast and patching services - implementation, measurement, and analysis in an internet streaming video testbed." To appear in *ACM Multimedia system 2001*.