PROXY SUPPORT FOR
STREAMING ON THE INTERNET

Layer-Encoded Video Streaming: A Proxy's Perspective

Michael Zink, University of Massachusetts
Jens Schmitt, University of Kaiserslautern
Carsten Griwodz, University of Oslo

ABSTRACT

This article describes a proxy's perspective in an architecture that supports efficient distribution of recorded video data in today's Internet. The support of layer-encoded video streaming with the aid of proxies enables a video distribution infrastructure that is efficient in *today's* Internet, and allows new mechanisms and techniques to be leveraged in a future Internet. An overview of a scalable and adaptive streaming architecture is presented. Open problems in the distribution and caching process are identified. Throughout the article solutions to improve the performance of such a video distribution architecture are discussed, mainly from the perspective of a proxy cache. The investigations that led to these solutions are based on an empirical experiment with layer encoded video. The goal of this experiment was to understand the effect of varying the number of layers on the viewer's perceived quality. The results of this initial investigation can be seen as the foundation for subsequent studies on mechanisms that improve the transport and caching of layer-encoded video in a scalable adaptive streaming architecture.

INTRODUCTION

FACTS OF INTERNET LIFE

Several investigations on streaming media in the Internet have shown that video streaming is becoming more and more popular,[1] and it is very likely that this increase in popularity continues in the near future.

Although there has been an immense amount of work on quality of service (QoS), the only service currently offered in the entire Internet is best effort. This means resources in the Internet cannot be reserved; thus, no guarantees for a certain service (e.g., a guaranteed amount of bandwidth, packet loss, and latency on the link between two nodes) can be offered. The congestion control mechanisms in TCP ensure that each session receives its fair share of the available bandwidth. Since UDP does not provide a congestion control mechanism, a UDP-based transmission consumes as much bandwidth as available, with the consequence that TCP-based transmissions do not get their fair share of the bandwidth. This phenomenon is described as TCP-unfairness, while transmission protocols that behave like TCP are called *TCP-friendly* (resulting in TCP-fairness). Being good Internet "citizens," media streaming applications should be designed to be TCP-friendly.

Recent developments in the end system market increased the *heterogeneity* of end systems (phones, game consoles, 64-bit desktops) and access links (general packet radio service, GPRS, vs. 8 Mb/s digital subscriber line, DSL) used by those systems. Therefore, a video distribution architecture that fits well in today's Internet should be developed with consideration of two major guidelines. First, to overcome the rising problem of TCP-unfairness, streaming should be performed in a congestion controlled manner. Second, an adaptation of the video stream to a wide spectrum of access link capacities and end system characteristics should be possible.

SCALABLE VIDEO DISTRIBUTION

Delivering video streams to users in a scalable fashion can hardly be thought of without a distribution infrastructure that allows these still very resource-intensive requests to be served close to where they originated. In effect, this means cache proxies are an important ingredient of successful video distribution systems.

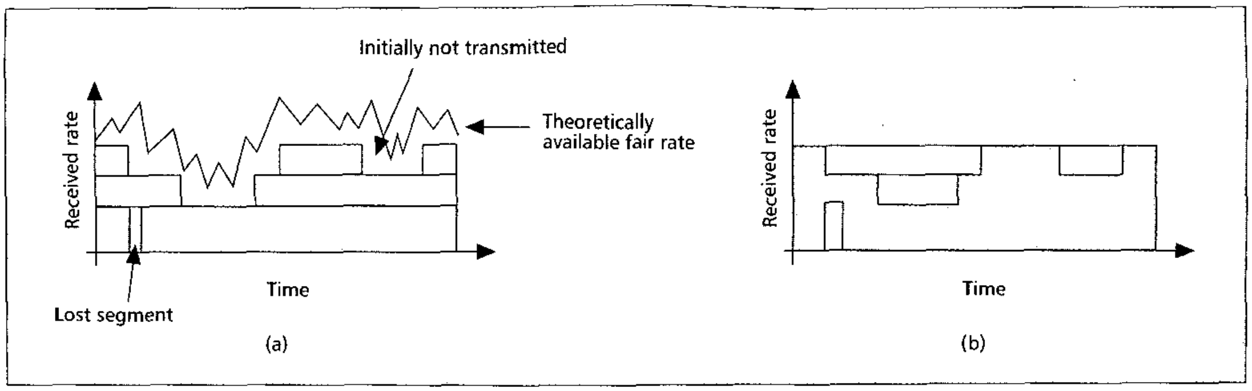
Investigations of video distribution systems have shown the following typical traits:

- Most video streaming applications follow the write-once read-many principle; hence, cache consistency is not a major issue.
- Accesses to videos exhibit strong temporal locality. This means it is highly probable that an object accessed recently with high frequency will be requested again soon.

Michael Zink was a member of the Multimedia Communications Laboratory at TU Darmstadt while the work presented in this article was developed.

[1] *http://www.streaming-media.com/research, Current Reports, "Streaming Media Growth Analysis & Market Dynamics: 2002 vs. 2001," Executive Summary.*

0163-6804/04/$20.00 © 2004 IEEE



□ **Figure 1.** *a) Cached layer-encoded video; b) retransmitted segments.*

- Investigations of the popularity of video objects have shown that approximately 80 percent of user requests are concentrated on 20 percent of the total amount of available videos ([1, 2]).

These results indicate that caching can be a very effective means to achieve scalability of video distribution systems.

With the aforementioned benefits of caching in video distribution systems and the necessity of adaptive streaming in mind, a scalable adaptive streaming architecture was designed with proxy caches as a focal building block of its infrastructure.

Scalable Adaptive Streaming

System Scalability

Let us briefly describe our video caching architecture. As our caching method we employ so-called *write-through caching*,[2] where a requested stream is forwarded through the proxy cache if the cache replacement strategy decides to store the requested video on the proxy cache. Subsequent clients can then be served from the proxy cache. Write-through caching requires a mechanism to recover from packet losses between server and proxy cache. Our mechanism, called Loss Collection Real-Time Transport Protocol (LC-RTP) [3], fits well in a video distribution architecture that employs write-through caching (an alternative is, e.g., [4]). In contrast to TCP, the time and amount of retransmissions in LC-RTP are controlled by the application. For example, retransmissions can be initiated *after* the actual transmission and not by the transmission protocol itself *during* a transmission (as for TCP).

Content Scalability

In order to enable congestion control for inelastic applications like video streaming, a quality adaptation mechanism is needed. The visual quality of streams decoded from non-scalable formats like MPEG-1 degrades quickly with increasing data loss. This is different if scalable encoding schemes are applied. In this case, part of the video data is sufficient to reconstruct the video signal with the trade-off that the quality of this video signal is reduced. An encoding scheme that makes use of scalability is layered encoding. Layered encoding splits the video into *segments* along the time axis like classical video encoding (at the granularity of frames or groups of pictures), but in addition splits temporal segments into one *base layer* and several *enhancement layers*. The base layer contains fundamental coding information and can be decoded without any additional information. Enhancement layers contain additional information that increases the quality of the reconstructed video signal. In contrast to the base layer, enhancement layers are not independent of other layers. To reconstruct the information included in layer n all of the information on the lower layers $(0, ..., n - 1)$ is required. We consider the case of hierarchical layer encodings that support only a small number of relatively big rate steps. The fine-grained scalability (FGS) approach [5] uses one mandatory base layer and enhancement information that can be cut off at an arbitrary byte, leading to a large number of small rate steps.

Besides the basic argument for applying content scalability, its varying transmission rate, the heterogeneity of clients and access networks leads to various transmission, processing, and display capabilities such that different clients have very different demands with respect to the quality of the transmission. Scalable content allows these different demands to be served using the same source file.

Combining System and Content Scalability

Figure 1a shows a possible version of a cached layer-encoded video when a TCP-friendly video transmission is combined with write-through caching. The figure shows a base layer with a gap that invalidates the enhancement layers and two partially received enhancement layers. Obviously, the cached copy of the video exhibits a potentially large amount of missing data from different layers. Note that the exact shape of cached video content is a function of the congestion control mechanism being applied by the chosen TCP-friendly transmission protocol.

Clients that request the same content at a later point in time and would be served from the proxy[3] have no chance to receive the video in full quality if no extra measures are taken. Thus, a mechanism is required that improves the quality of the cached content. Figure 1b depicts the

[2] *This terminology is adopted from memory hierarchies.*

[3] *In the remainder of this article the terms* proxy *and* proxy cache *are used interchangeably.*

#	Clip	Farm1		Farm2		M&C1		M&C3		M&C4		T-Tennis3	
	Metric	ts 1	ts 2	ts 1	ts 2	ts 1	ts 2	ts 1	ts 2	ts 1	ts 2	ts 1	ts 2
1	Subjective assessment (> 0 if ts 2 better)	0.35		0.55		0.73		1.18		1.02		2.18	
2	PSNR (higher is better)	62.86	49.47	61.46	73.28	63.15	52.38	48.01	25.08	49.40	26.95	66.02	63.28
3	Spectrum (lower is better)	2	2	6.86	4	2	1	2	0	2	0	0.5	0.5

Pink: contrary to subjective assessment
Blue: in accordance with subjective assessment
Light blue: inconclusive

Table 1. *A comparison among subjective quality, PSNR and spectrum (ts: test sequence).*

> *One basic conclusion from the results of the subjective assessment is: adding information to a layer-encoded video increases its average quality eventually, but not monotonically. Yet, adding information at different locations can have a substantial effect on the perceived quality.*

parts that would be identified by such a mechanism and transmitted from the server to the proxy, leading to a full-quality copy of the video object.

In the following, we use the term *retransmission* for all transmissions of missing data from the server to the proxy caused by the proxy's request. This definition might seem confusing because some of the missing data may have never been transmitted at all, but the proxy cannot distinguish between packets that were not sent at all (initially not transmitted) and lost packets. Thus, for the proxy every packet that was not transmitted initially appears as a retransmitted packet.

Quality Variations in Layer-Encoded Video

The drawback of adaptive layer-encoded video transmissions is the introduction of variations in the number of transmitted layers (i.e., layer variations) during a streaming session. These variations affect the end user's perceived quality and thus the acceptance of a service based on such technology. Hence, all transport mechanisms that try to optimize transmission should base their decisions on perceived quality or metrics for it. Extensive literature research revealed a lack of in-depth analysis of the influence of layer variations on the viewer's perceived quality. Thus, an empirical experiment was conducted that involved subjective assessment to obtain results that can be applied to classify the perceived quality of such videos.

Subjective Assessment Results — The main goal of this investigation was to identify the relation between objective quality metrics and subjective quality. Assumptions on the quality of layer-encoded video are that the quality is affected by not only the total sum of received segments but also the frequency of layer variations and the amplitude of those variations. To acknowledge or refute these assumptions a subjective assessment with more than 100 test candidates was performed according to the International Telecommunication Union (ITU) standard for video assessment [6]. A detailed description of the subjective assessment can be found in [7].

A statistical analysis of the experiment in large parts validates assumptions made about layer variations and the perceived quality of a video:

- The frequency of variations should be kept as small as possible.
- If variation cannot be avoided, its amplitude should be kept as small as possible.

One basic conclusion from the results of the subjective assessment is that adding information to a layer-encoded video increases its average quality eventually, but not monotonically. However, adding information at different locations can have a substantial effect on the perceived quality. Thus, it is more likely that the perceived quality of a layer-encoded video is improved if:

- The lowest quality level is increased.
- Gaps (i.e., ranges of consecutively missing segments for one layer) in lower layers are filled.

Our findings imply that although FGS would be able to adapt to changes in available bandwidth much better than a coarse-grained layering scheme, using it without prefetching or other smoothing techniques would have negative effects on the perceived quality because of a large number of quality changes.

Objective Quality Metrics vs. Subjective Quality — Peak signal-to-noise ratio (PSNR) is a popular metric to present the objective quality of video data. Therefore, the average PSNR for the sequences used in the subjective assessment was determined and compared to the results of the latter. This comparison revealed that PSNR is not an adequate metric to represent the influence of variations in layer-encoded video on perceived quality. Rows 1 and 2 in Table 1 provide a representative example. Six video clips are encoded twice (test sequences 1 and 2) with layer changes at different points in

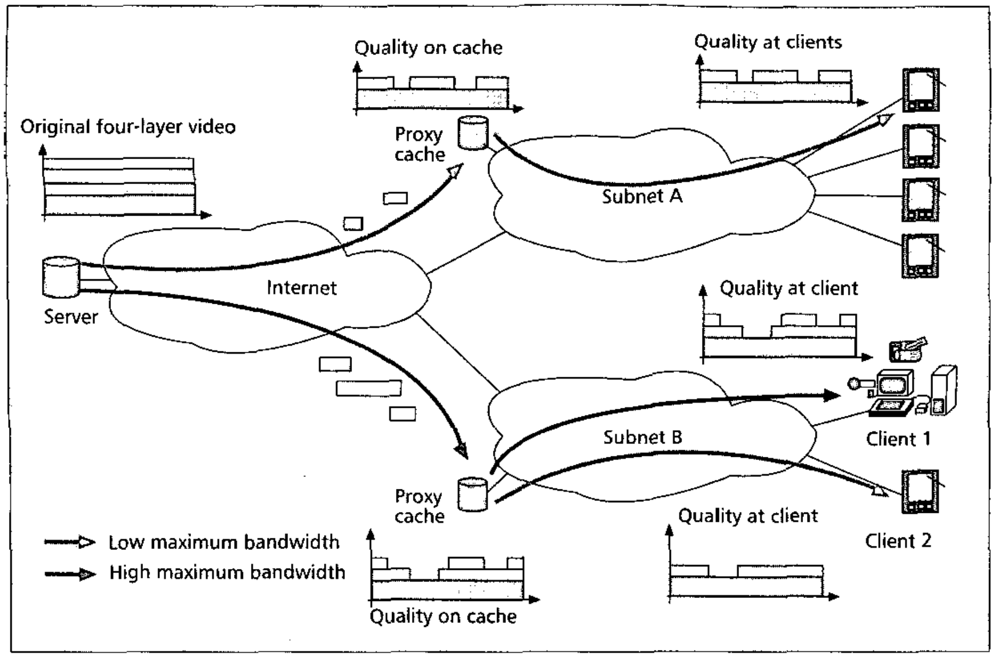


Figure 2. *A scalable video distribution system for heterogeneous clients*

Although FGS would be able to adapt to changes in available bandwidth much better than a coarse-grained layering scheme, using it without prefetching or other smoothing techniques would have negative effects on the perceived quality because of a large number of quality changes.

time. According to the subjective evaluation in row 1, ts 2 has the better subjective quality in all cases. Row 2 shows the PSNR values of test sequences, and for five of the six video clips the results are contrary to those of the subjective assessment. Thus, for layer-encoded video the quality of a sequence is not well represented by the PSNR.

This lack of an appropriate objective quality metric for layer-encoded video led to a new metric called *spectrum*. The goal during the development of spectrum was to express the factors that influence the perceived quality of a layer-encoded video through a mathematical expression giving similar results to those obtained by the subjective assessment. Therefore, the spectrum of a cached layer-encoded video v can be introduced:

$$s(v)=\sum_{t=1}^{T} z_t\left(h_t-\frac{1}{\sum_{i=1}^{T} z_i}\left(\sum_{j=1}^{T} z_j h_j\right)\right)^2 \qquad (1)$$

with h_t and z_t defined as:

- h_t — number of decoded layers in time slot $t, t = 1, \ldots, T$
- z_t — indication of a step $(h_t \neq h_{t-1})$ in time slot t, $z_t \in \{0,1\}$, $t = 1, \ldots, T$

The spectrum captures the frequency as well as the amplitude of layer variations. The amplitude is captured by the differences between quality levels (h_t) and average quality levels where larger amplitudes are given higher weight due to squaring these differences. The frequency of variations is captured by z_t. Only those differences that correspond to a step in the cached layer-encoded video are taken into account. A spectrum of value 0 represents the best possible quality, while the spectrum increases with decreasing quality.

Note that our new metric does not yet capture that lower layers are more relevant than higher layers, or when the layer changes occur during the course of a video clip playout, but a comparison of the spectrum of test sequences with the average result of the subjective assessment and PSNR shows that spectrum is already a more suitable objective metric than PSNR. In our example in Table 1, we see that the spectrum in row 3 yields the correct indication for four of the six clips, inconclusive in two cases, and no incorrect indication. In addition, online calculation of a metric can be performed in a simpler way with spectrum than with PSNR.

A Scenario for Scalable Adaptive Streaming

To clarify why it is important to combine system and content scalability in a video distribution system that is well-suited for today's and the future Internet, a scenario that uses the scalable adaptive streaming architecture is presented in this section.

The scenario shown in Fig. 2 depicts a heterogeneous distribution system consisting of two subnets connected to the Internet backbone. In each of these subnets a proxy cache is located to which all client requests are directed. Subnet A has a wireless infrastructure in which only homogeneous clients (in terms of link bandwidth) are connected, while subnet B has a heterogeneous infrastructure. Let us assume that the original video objects stored at the server consist of four layers. In subnet A none of the clients is able to receive more than two layers. In subnet B the content might be first requested by a handheld device (client 2) at lower quality due to its restricted access bandwidth. A subsequent client

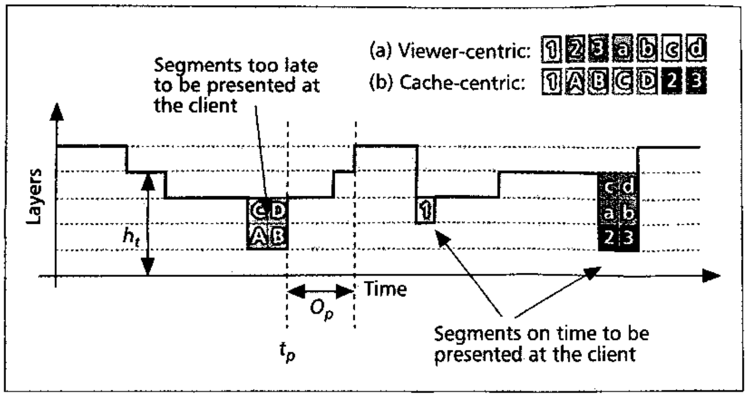

(a) Viewer-centric: 1 2 3 a b c d
(b) Cache-centric: 1 A B C D 2 3
Segments too late to be presented at the client
Layers
h_t
C D
A B
1
c d
a b
2 3
O_p
t_p
Time
Segments on time to be presented at the client

Figure 3. *Viewer-centric and cache-centric U-SG-LLF operation.*

requesting this content is a high-end PC that would like to receive the content at better quality and has an access link with high bandwidth characteristics. It benefits from additional transmissions from the server to the proxy cache that improve the quality of the cached content. Depending on the order in which client 1 and client 2 request the video, it might be that client 1 receives a stream from the proxy cache that has reduced quality because no higher quality is available at the proxy cache. This can be for two reasons. Either the stream was originally requested by client 2, which did not request more than two layers, or the cache replacement strategy might have dropped the upper two layers due to space constraints. If a retransmission mechanism is used, this situation can be circumvented. After initially delivering the stream to client 2, the proxy cache starts to request missing data of certain layers upon request of client 1.

Retransmission Scheduling

Retransmission scheduling is a method that improves the quality of a layer-encoded video that has been cached incompletely and is requested by a client. In Fig. 2 we see how only part of the video object is initially stored in each proxy cache. A retransmission process can be used to complete this video object when the object is requested by another client that is served by the same proxy at a later point in time. Then the proxy requests missing segments and forwards them in combination with the segments of the cached video object to the client. Thus, the quality of the video object streamed to the client has higher quality than the initially cached object. Depending on the decision of the proxy, the retransmitted segments can also be stored on its cache in order to increase the quality of the cached object.[4]

Since an investigation of optimal retransmission scheduling showed it cannot be applied in scalable adaptive streaming due to the fact that it is computationally too expensive [8], heuristics are used that have much lower complexity. The most interesting issue here is how to schedule the retransmissions; that is, in which order to retransmit missing segments in order to achieve certain quality goals for the streamed/cached video content.

A Heuristic for Retransmission Scheduling

In this section a heuristic is introduced as a representative of a family of heuristics for retransmission scheduling. In an investigation based on simulations this heuristic performed best and is therefore presented here.

The heuristic is aimed at adapting scheduling decisions to the results obtained from the subjective assessment. This is in contrast to an approach [9] in which heuristics are not based on the outcome of a subjective assessment.

The heuristic is based on prioritization of the missing segments aimed at closing gaps. This prioritization is achieved by first sorting the segments according to the length of the gap they belong to and then using their layer levels as a second-order sorting criterion. The resulting heuristic is called Unrestricted Shortest Gap Lowest Layer First (U-SG-LLF). An example of the scheduling of missing segments is given in Fig. 3.

Retransmission Focus

Retransmission scheduling can be subdivided into two types that have the goal to maximize the quality for either the current viewer or the cached video. The first type, described as *viewer-centric*, is shown in Fig. 3a in the sequence of retransmitted segments. To ensure that the retransmitted segments do not arrive after their playout time (t_p) to the current client, a prefetching offset O_p is introduced. O_p should be chosen sufficiently large such that O_p > round-trip time (RTT) for the transmission path between server and proxy at all times. A drawback of the viewer-centric type is the fact that due to the offset O_p, missing segments at the beginning of the video will never be requested for retransmission. Alternatively, a second type called *cache-centric* can be used. With this modification segments for which the playout time t_p has already passed can also be scheduled (Fig. 3b). Thus, the overall

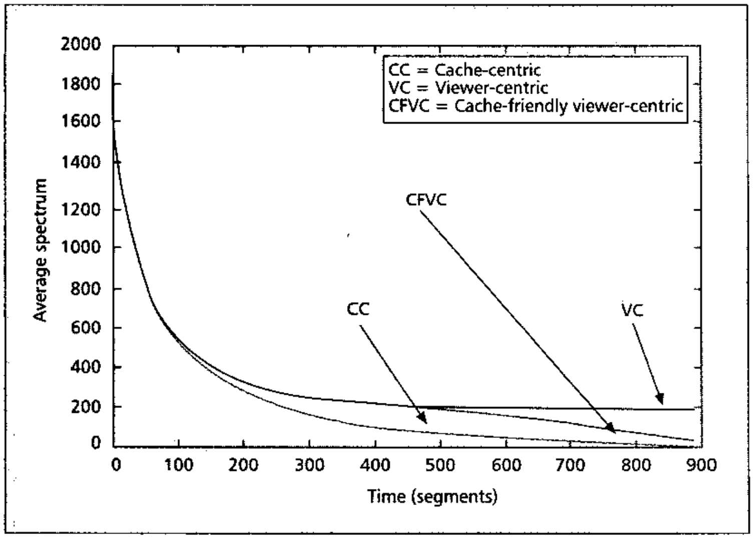


Figure 4. *Average spectrum of 1000 simulation runs for each heuristic (10 layers, retransmission bandwidth = 4)*

[4] *In the remainder of this article it is assumed that the proxy always decides to cache retransmitted segments.*

quality of the cached video is improved for all subsequent potential viewers, not only the current one.

One special case that can occur with viewer-centric retransmission scheduling is when no more segments are retransmitted although there are still missing segments. This occurs when all missing segments with a playout time larger than t_p have been retransmitted. At this point in time there are no further segments that have a playout time larger than t_p. This means that all remaining missing segments are useless at the client currently being served, and retransmitting any of them can only improve the quality of the cached content. Thus, they should not be forwarded from the proxy to the client. This type is called *cache-friendly viewer-centric* retransmission scheduling.

Simulation Results

In order to compare the different retransmission scheduling types simulations were performed [8]. Some of the results of these simulations are presented in Fig. 4. The cache-friendly viewer-centric type results in a better spectrum on the proxy cache than the plain viewer-centric type. On the other hand, the resulting spectrum for the cache-friendly viewer-centric type is slightly higher than the cache-centric type. Note that the resulting spectrum for the cache-friendly viewer-centric type at the client is identical to the spectrum of the viewer-centric type. Altogether, the simulation results reveal that retransmission scheduling (independent of the type) strongly improves the quality of a cached layer-encoded video, shown by the reduction of the spectrum. Fine-tuning can be performed by the proxy administrator by choosing the appropriate retransmission scheduling type.

Fair Share Claiming

Transmitting a layer-encoded video in a TCP-friendly manner does not always result in the session claiming its fair share of network resources. A change in the actual transmission rate might not necessarily result in a rate change for the layer-encoded video because the encoding format provides only a discrete number of different layers resulting in a finite number of possible transmission rates. This implies it is likely that the actual transmission rate is higher than the rate needed for the transmission of one or two layers at some points in time. This additional bandwidth is the fair share that may be claimed by a corresponding TCP session, yet due to the discrete nature of layer-encoded video it will not be claimed. Nevertheless, finding some data to fill this gap would allow the stream to claim its fair share without breaking the cooperative rules implied by TCP's resource allocation model. Thus, we call this method *fair share claiming* (FSC). An obvious use of this additional bandwidth could be to transmit segments identified by retransmission scheduling and thereby claim the fair share for a TCP-friendly streaming session. Also, transmissions of segments from other videos on this proxy could be performed, even transmission of segments of currently running streaming sessions could be considered. However, the latter comes at the price of some implementation complexity, because these segments must be sent over the TCP-friendly streaming session that currently underutilizes its fair share. Note that without FSC this multiplexing between streams from the same server is not possible in an environment like the Internet where many other flows coexist at the bottleneck link between origin server and proxy.

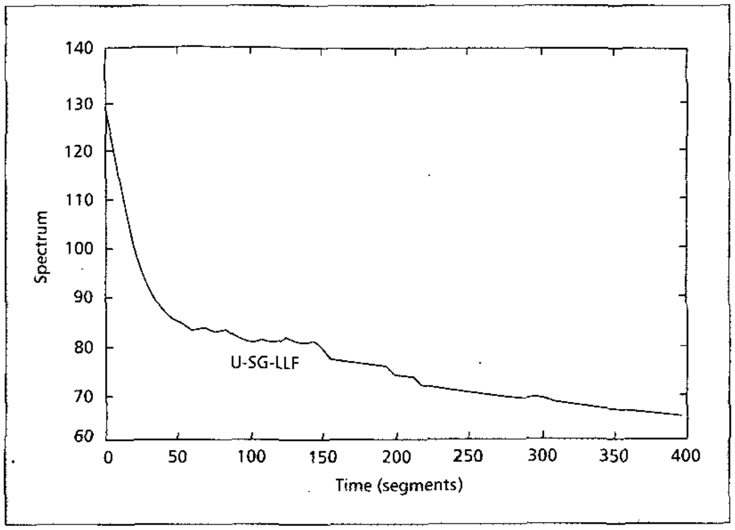


Figure 5. *Spectrum for cache-centric retransmission scheduling with FSC.*

Figure 5 shows the result of the simulation for the cache-centric retransmission scheduling presented earlier. The outcome of this simulation confirms the results of the simulation presented previously where a constant bandwidth available for retransmissions during the whole simulation was assumed. An interesting detail is that the spectrum is not monotonically decreasing. This is caused by the fact that in some cases only a small amount of bandwidth is available for retransmissions, so gaps will not be closed completely or, even worse, segments of layers that were not cached at all are retransmitted. The latter increases the amount of layer changes, which leads to increased spectrum. This short-term increase should be accepted to allow quality improvement of cached video in the long run.

Additionally, the results of the simulations show that fair share claiming is a valid method of performing retransmissions from servers into proxy caches without affecting the quality of the transported video stream.

Polishing

In this section a technique called *polishing* is presented. With polishing a proxy cache considers sending only a subset of the segments of a locally stored object in order to reduce layer variations at the client. Our investigations on the perceived quality of layer variations in videos mentioned earlier show that it can be beneficial to omit the transmission of certain segments, especially if the amount of layer variations is reduced.

At first, this might sound counterintuitive

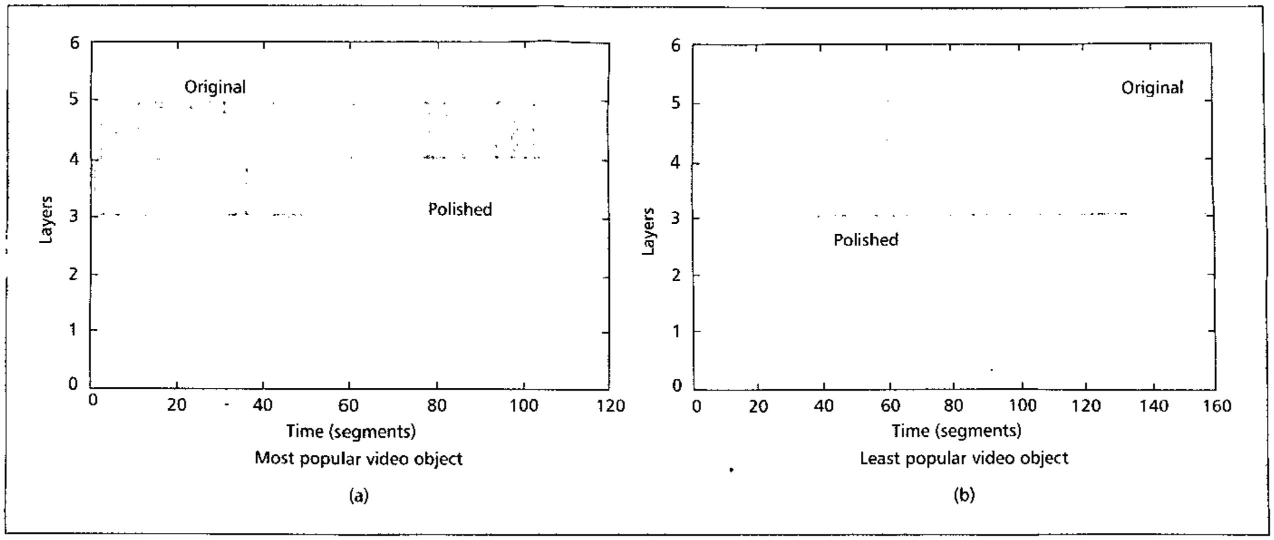


Figure 6. *Polishing for most and least popular video objects.*

since some information is not transmitted at all, and thus the PSNR of the video is reduced. However, our investigations have shown that despite reducing the PSNR, reducing layer variations can increase the perceived quality of a video, if done carefully.

Polishing can be applied when the storage space at the local cache is exhausted and new video objects should be stored on the cache. Instead of removing complete less popular objects from the cache, polishing can be applied to remove a certain number of segments. Thus, new objects can be added to the cache's storage, while only segments of certain objects are removed instead of complete objects.

Popularity-Based Cache Replacement

In [10] a detailed description of the investigations on polishing is given. It was the goal to investigate whether polishing can be applied to cached videos based on their popularity. This would mean fewer segments deleted from popular videos while the number of deleted segments increases for less popular objects. Thus, the quality of the cached object is directly related to its popularity. In this case the complete content of the cache is regarded and polished according to the popularity of each single object and the amount of space that should be freed. In the following we consider only a replacement policy that maintains a low spectrum for cache content, even though a popularity-based combination with retransmission scheduling is also conceivable.

The Popularity-Based Cache Replacement Simulation

A simulation was performed to demonstrate the applicability of polishing as a popularity-based cache replacement algorithm. The amount of cache space that should become available for the caching of new data, $K^{total} - K^{max}$, can be specified for the simulation. In addition, each video object is assigned a certain popularity. Figure 6 shows the originally cached and resulting polished video object for the video with the highest (a) and lowest (b) popularity on the cache. In this case, 10 video objects are stored on the cache and 25 percent of the total cache space is freed by polishing the cached videos according to their popularity. For the most popular video object 20 percent of the original segments are deleted, while for the least popular video object 31 percent are removed from the cache.

Table 2 shows the number of segments (percent) removed from each of the 10 cached video objects. Objects that are shaded equally were assigned the same popularity value. The popularity is highest for objects 1, 2, and 3, and lowest for objects 7, 8, 9, and 10. The popularity for 4, 5, and 6 lies between the other two groups. The results of this simulation show that with the polishing algorithm a very fine-grained cache replacement can be achieved. With this algorithm it is possible to free cache space for new content, while data from already cached content is removed according to the popularity of the content. In this specific example 25 percent of

Video object	1	2	3	4	5	6	7	8	9	10
Average number of removed segments (%)	18	14	18	26	26	25	32	31	32	31
Average Spectrum of polished object	5.6	4.3	5.3	3.1	4.0	5.4	2.2	2.0	2.8	2.4
Average Spectrum of unpolished object	18.1	19.3	19.9	23.3	32.2	26.7	23.8	21.3	23.9	25.8

Table 2. *Average number of removed segments.*

the caches' storage space is available for the caching of new content, while none of the cached objects had to be removed completely.

CONCLUSIONS

In this article we first present the spectrum of layer-encoded video, a new metric for the objective quality of such video. Through a subjective assessment we give evidence that this metric is superior to the PSNR metric commonly applied. We subsequently demonstrate how the spectrum can be applied to decision problems in video distribution with proxy caches.

One application is retransmission to enhance the quality of a layer-encoded video stored in a proxy cache. In this article we distinguish viewer-centric and cache-centric retransmission scheduling types. The viewer-centric type provides higher quality to the requesting client, but does not achieve the best quality for the proxy cache's copy. The cache-centric type improves the quality of a video on the proxy but reduces the experienced quality of the first requesting clients. We identify a third approach, cache-friendly viewer-centric, that initially follows the viewer-centric approach but achieves almost perfect video quality on the proxy by later switching to the cache-centric type.

We further improved our retransmission approach by adding fair share claiming. In this approach we observe that TCP-friendly real-time streaming of layer-encoded video rarely consumes exactly the bandwidth the TCP-friendly algorithm allows. We show that our fair share claiming approach can use this bandwidth to quickly improve the quality of a layer-encoded video in the proxy cache according to the spectrum during an ongoing transmission.

Finally, we show that our metric can likewise be applied to popularity-based cache replacement decisions. The polishing technique is a means of removing segments of higher layers from a proxy cache in response to a replacement decision such that the remaining quality is the highest possible.

The results of this simulation show that with the polishing algorithm a very fine-grained cache replacement can be achieved. With this algorithm it is possible to free cache space for new content, while data from already cached content is removed according to the popularity of the content.

REFERENCES

[1] G. Bianchi and R. Melen, "Non Stationary Request Distribution in Video-on-Demand Networks," *Proc. INFOCOM '97*, Kobe, Japan, Apr. 1997, pp. 711–17.

[2] C. Griwodz, M. Bär, and L. C. Wolf, "Long-Term Movie Popularity in Video-on-Demand Systems," *Proc. ACM Multimedia Conf. 1997*, Seattle, WA, Nov. 1997, pp. 340–57.

[3] M. Zink et al., "LC-RTP (Loss Collection RTP): Reliability for Video Caching in the Internet," *Proc. 7th Int'l. Conf. Parallel and Distrib. Sys.: Workshops*, Iwate, Japan, July 2000, pp. 281–86.

[4] M. Mauve *et al.*, "RTP/I — Toward a Common Application Level Protocol for Distributed Interactive Media," *IEEE Trans. Multimedia*, vol. 3, no. 1, Mar. 2001, pp. 152–61.

[5] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*, Prentice-Hall, 2002.

[6] ITU-R BT.500-10, "Methodology for the Subjective Assessment of the Quality of Television Picture," 2000.

[7] M. Zink *et al.*, "Subjective Impression of Variations in Layer Encoded Videos," *Proc. IWQoS '03*, Monterey, CA, June 2003, pp. 134–54.

[8] M. Zink, J. Schmitt, and R. Steinmetz, "Retransmission Scheduling in Layered Video Caches," *Proc. ICC '02*, New York, NY, Apr. 2002, pp. 2474–78.

[9] R. Rejaie et al., "Multimedia Proxy Caching for Quality Adaptive Streaming Applications in the Internet," *Proc. INFOCOM 2000*, Tel-Aviv, Israel, Mar. 2000, pp. 980–89.

[10] M. Zink *et al.*, "Polishing: A Technique to Reduce Variations in Cached Layer-Encoded Video," *MMCN 2004*, San Jose, CA, Jan. 2004, pp. 187–98.

BIOGRAPHIES

MICHAEL ZINK (zink@cs.umass.edu) is currently a postdoctoral fellow in the Computer Science Department at the University of Massachusetts in Amherst. Previously he was a researcher at the Multimedia Communications Laboratory at Darmstadt University of Technology. He works in the fields of sensor networks and distribution networks for high bandwidth data. Further research interests are in wide-area multimedia distribution for wired and wireless environments and network protocols. He is one of the developers of the KOMSSYS streaming platform. He received his Diploma (M.Sc.) from Darmstadt University of Technology in 1997. From 1997 to 1998 he was employed as guest researcher at the National Institute of Standards and Technology (NIST) in the United States, where he developed an MPLS testbed. In 2003 he received his Ph.D. degree (Dr.-Ing.) from Darmstadt University of Technology; his thesis was on scalable Internet video-on-demand systems.

JENS SCHMITT (jschmitt@informatik.uni-kl.de) is a professor in the Computer Science Department at the University of Kaiserslautern where he heads the Distributed Computer Systems Laboratory (DISCO). Previously, he was research group leader of the Multimedia Distribution & Networking group in the Multimedia Communications Laboratory (KOM) at Darmstadt University of Technology. He works in the fields of QoS provisioning in distributed systems, in particular in heterogeneous network scenarios, QoS for mobile communications, and scalable distribution of multimedia content with an emphasis on high availability systems. Further research interests are in network traffic modeling, real-time scheduling, and evolutionary algorithms. Jens Schmitt received his Master's degree (Dipl.) from the University of Mannheim in joint business and computer sciences in 1996. In 1994, during a stay at the University of Wales, Swansea, he also did a European M.B.S. degree. In 2000 he received his Ph.D. degree (Dr.-Ing.) from Darmstadt University of Technology; his thesis was on heterogeneous network QoS systems.

CARSTEN GRIWODZ (griff@ifi.uio.no) is an associate professor in the Department of Informatics at the University of Oslo, Norway. He joined the Distributed Multimedia Systems group in Oslo in 2000. He works on issues of wide-area scalability of applications ranging from media on demand systems to massive multiplayer games, and on middleware for distributed applications in ad hoc networks. His main research interest is the improvement of mechanisms and algorithms for media servers, interactive multimedia, and distribution systems. He received his Master's (Dipl.) degree in computer science from the University of Paderborn, Germany, in 1993. From 1993 to 1997 he worked at the IBM European Networking Center, Heidelberg, Germany. In 1997 he joined the Multimedia Communications Laboratory at Darmstadt University of Technology, Germany, where he received his Ph.D. (Dr.-Ing.) degree in 2000.