Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 1136–1141
computer communications
www.elsevier.com/locate/comcom

Convergence of periodic broadcasting and video-on-demand ☆

Juan Segarra [a,b,*], Vicent Cholvi [c]

[a] *Dpt. Informática e Ingeniería de Sistemas and Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain*
[b] *European HiPEAC Network of Excellence*
[c] *Dpt. Lenguajes y Sistemas Informáticos, Universitat Jaume I, Spain*

Received 31 August 2006; received in revised form 29 November 2006; accepted 4 December 2006
Available online 2 January 2007

Abstract

Research on video-on-demand transmissions is essentially divided into periodic broadcasting methods and on-demand methods. Periodic broadcasting is aimed to schedule transmissions off-line, so that an optimized time schedule is achieved. On the other hand video-on-demand has to deal with constraints at requesting times. Thus, studies on these areas have been quite isolated. Obviously, in periodic broadcasting all parameters are known in advance, so timetables can be accurately adjusted and it is assumed transmissions can be arranged to use less bandwidth than video-on-demand.

In this paper, we analyze the convergence of both paradigms, showing that the claims that argue that VoD schemes use more bandwidth than PB ones are not necessarily true. We state this argument by proving how to convert any periodic broadcasting method into an on-demand one, which will use equal or less bandwidth. Moreover, we show that this converted on-demand method can also offer shorter serving times.

Keywords: Video-on-demand; Periodic broadcasting; Broadcast; Multimedia

1. Introduction

Studies on multimedia transmissions offering quality of service (i.e. guaranteeing that when users start to watch the video it will be played without interruption and loss of quality) have been mainly addressed from two different perspectives. Whereas *video-on-demand* (VoD) schemes provide users with the possibility of watching videos at any time, *periodic broadcasting* (PB) schemes are used to maximize the throughput of transmissions, behaving as a cinema timetable where each movie has a predefined playing/transmission time. In this way, PB schemes sacrifice the "on-demand" behavior to obtain an off-line optimized scheduling, whereas VoD ones need an on-line scheduler and are dependent on concrete requests.

1.1. Video-on-demand

VoD systems appeared as a step from typical transmissions, i.e. a best effort server offering several videos, and occasionally grouping several requests of the same video for their transmission [3]. Following this initial stage, schemes considering QoS appeared. The *piggybacking* technique [10] consists of varying the transmission rate of the streams in order to join streams of the same video that have been requested at different times. The *tapping* method [6] allows users to receive (and store in buffers) partial streams currently being served to other users. The *patching* technique [5] is a similar approach which divides streams into independent segments. In this way transmissions can be planned more efficiently because the transmission of each segment can be closely adjusted to its playing deadline.

☆ This work was partially supported by the MEC under Grants No. TSI2004-02940 and TIN2004-07739-C02-02 (50% ERDF/FEDER), by Bancaixa under Grant No. P1-1B2003-37 and by the DGA under Grant "Grupo Consolidado de Investigación".
* Corresponding author. Tel.: +34976762109.
E-mail addresses: jsegarra@unizar.es (J. Segarra), vcholvi@uji.es (V. Cholvi).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.12.007

Additionally, transmitting several segments at the same time is also easier using predefined video units [16]. The *stream merging* technique [8] uses an approach that considers videos as continuous streams, although it plans their transmission considering time units. In this way it can schedule transmission depending both on the expected requests for each video and their current transmission status.

1.2. Periodic broadcasting

Perhaps the most simple periodic broadcasting scheme is to retransmit the same video time after time on a channel. The main objective of these methods is the minimization of bandwidth by maximizing the receivers [2]. Thus, knowing that the most popular videos are far more requested, a transmission plan can be developed off-line to fulfill this goal. In order to carry out this maximization of receivers, all broadcasting schemes have reached the conclusion that, by dividing a stream into several small ones, it is easier for requests to arrive on time to receive several of these small segments. Thus, the first segments of movies must be small and broadcast very frequently, and the later ones, being less urgent, can be broadcast at a lower frequency and their size can be larger. Perhaps the most well known scheme is the *Pyramid Broadcasting* [19], which uses a geometrical segmentation. However, its drawback is that, depending on the time when videos are requested, clients must be able to receive many channels at the same time. This has been the main reason for other schemes to appear, such as the *Permutation-based Pyramid Broadcasting* [1], which rearranges the transmission slots of segments, or the *Skyscraper Broadcasting* protocol [11], which works with series of pairs of segments so that the number of channels to receive at the same time is much lower. Another scheme, the *Fast Broadcasting* [13] adds the capacity of supporting clients unable to store segments locally.

Finally, the *Harmonic Broadcasting* [12] is based on dividing videos in segments of equal size. In this way, the ith segment is entirely transmitted every i time-slots using a bandwidth of $\frac{b}{i}$, where b is the video CBR bandwidth. However, this scheme does not guarantee data being at the client on time, and several solutions have been proposed to solve this problem [14]. One of these solutions, the *Polyharmonic Broadcasting* [15], consists of delaying the video playing so that it will guarantee all segments have been entirely broadcast at their consumption time. This delay also allows to slightly reduce the broadcast frequency of segments and therefore they can be transmitted using a lower transmission bit-rate. Assuming CBR, this method has been found to be optimal in bandwidth usage [9].

1.3. Our work

In this paper, we show that any (including optimal) off-line schedulings of the periodic broadcasting (PB) methods can be improved by on-line schedulers in video-on-demand (VoD).

We propose a transformation that, applied on any PB scheme, generates a VoD scheme providing equal waiting times and equal or less bandwidth requirements. Moreover, we show that it may be also possible to obtain VoD schemes offering both lower waiting times and lower bandwidth consumption than their associated PB schemes.

The main advantage of this transformation is that, additionally of using less resources, the cost efficiency of transmissions is far less dependent on the number receivers, since the system adapts transmissions to current requests. This means we can relax the accuracy of estimations of expected requests, or even not use them in some cases.

2. Popularity considerations

There are several details which many of the previous studies do not consider. The most important in QoS systems is probably the popularity of videos. Although recent studies provide accurate distributions for specific situations [17], it is widely accepted that the probability of video requests follows the *Zipf's Law* [4]. Having N videos ordered by popularity, this law defines the probability of requesting the ith most popular one as

$$P_N(i) = \left(i^{\alpha} \times \sum_{j=1}^{N} \frac{1}{j^{\alpha}} \right)^{-1}$$

with $0 < \alpha \leqslant 1$. This gives an exponential probability of request, where the least popular videos are hardly requested. Having this into account, note that PB schemes assume all videos are being constantly requested. Fig. 1 shows the minimum required clients for a range of offered videos (with $\alpha = 0.271$ [7]). In this figure clients request one video per week, and a video is considered requested by having one request each 90 min on low request rate times (around 1/20 request rate [18]). Although having linear increments, notice that under these conditions, to make a cost-efficient PB system it is required a number of clients around four orders of magnitude larger than the number of videos.

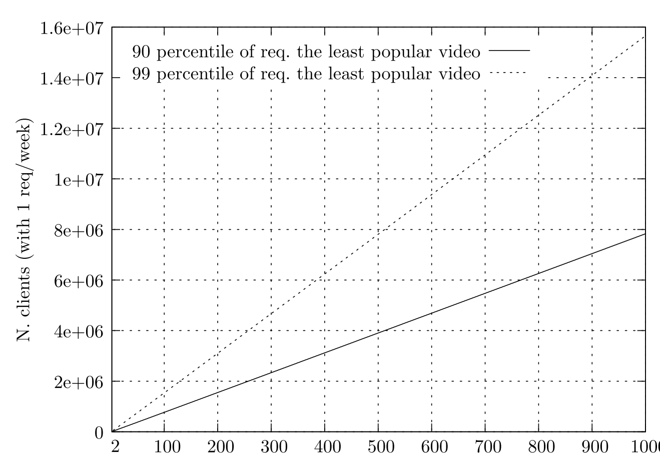

Fig. 1. Minimum required clients for a range of offered videos.

Also notice that a 99 percentile means that 1 out of 100 broadcast videos will have no receiver, which can be a significant waste of bandwidth. Depending on each concrete system, profitability may imply using other percentiles and/or having other factors into account.

3. Characterization of periodic broadcasting

Periodic broadcasting schemes retransmit video segments over time, and these transmission times are known, fixed and guaranteed in advance. Also, transmitting several segments at the same time in different channels is allowed. Since quality of service must be guaranteed, clients begin watching the video when the reception of all data before their deadline is assured (i.e. when the complete "visualization" of the video without interruption is guaranteed).

We can characterize these schemes by how they schedule their individual transmissions, since they are independent on requests. In this way, the complete scheduling system is the sum of all schedulings of each single video. At the same time, a video scheduling is usually composed of several transmissions of several video segments at different times. From the user point of view, given a video request, it would be served by a single transmission of each segment. Note that there are no restrictions about how segments are constructed, nor how their transmission times are set, since it depends on the PB scheme and will be specified by it.

Mathematically expressing these ideas, the transmission schedule using a given PB scheme of a video V consisting of n segments can be defined as a set of n tuples, as follows:

$$S(V,\mathrm{PB}) = \{\langle v_1, T_1\rangle, \langle v_2, T_2\rangle, \ldots, \langle v_n, T_n\rangle\}$$

where v_i identifies the ith segment of video V and T_i denotes the set of times (and bit-rates) when v_i starts to be transmitted. These times are specified by the scheduling system used by each concrete PB scheme.

Also, given a request R at time t of the video V, this request would be served by a single transmission of each segment in V:

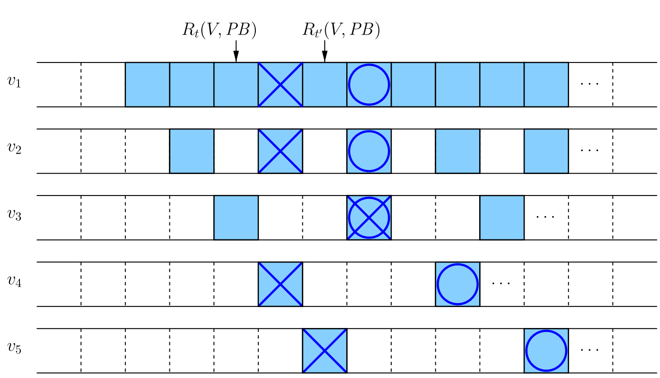

Fig. 2. Example of a simple *periodic broadcasting* scheme, broadcasting a video V with five segments, each one being transmitted by a different channel. It shows which segments would be received by requests $R_t(V,\mathrm{PB})$ (marked with ×) and $R_{t'}(V,PB)$ (marked with ○).

$$R_t(V,\mathrm{PB}) = \{\langle v_1, t_i \in T_1\rangle, \langle v_2, t_j \in T_2\rangle, \ldots, \langle v_n, t_k \in T_n\rangle\}$$

Fig. 2 shows an example of a simple PB scheme having two requests for the same video.

4. Convergence of PB and VoD

The main feature of VoD schemes is that they schedule transmissions on-demand, i.e. depending on the received requests. Given any periodic broadcasting method $S(V,\mathrm{PB})$, our goal is to obtain a VoD method $S(V,\mathrm{VoD})$ which follows the same scheduling policy but in an on-demand way. Since this transformed method schedules on-demand, it can be readily seen that the difference between $S(V,\mathrm{PB})$ and $S(V,\mathrm{VoD})$ is that, in $S(V,\mathrm{VoD})$, only the segments that are going to be "consumed" are scheduled and transmitted. From the user point of view there is no difference between both methods, since they schedule transmissions in the same way.

In order to make this transformation we propose the following rules:

(1) Start $S(V,\mathrm{VoD})$ with empty schedulings (i.e. $T'_i = \emptyset$ for all i):

$$S(V,\mathrm{VoD}) = \{\langle v_1, T'_1 = \emptyset\rangle, \langle v_2, T'_2 = \emptyset\rangle, \ldots, \langle v_n, T'_n = \emptyset\rangle\}$$

(2) For any request, take the n tuples $\langle v_i, t\rangle$ it would get using $S(V,\mathrm{PB})$, and update the corresponding tuples $\langle v_i, T'_i\rangle$ in $S(V,\mathrm{VoD})$ as follows:

$$\langle v_i, T'_i\rangle \leftarrow \langle v_i, T'_i \cup \{t \in T_i\}\rangle$$

Fig. 3 shows an example of how our proposed VoD scheme applies to the PB scheme in Fig. 2.

In the next proposition we demonstrate that our transformation offers equal waiting times and uses equal or less bandwidth.

Proposition 4.1. *Any PB scheme can be transformed into a VoD scheme which offers equal waiting times and uses equal or less bandwidth.*

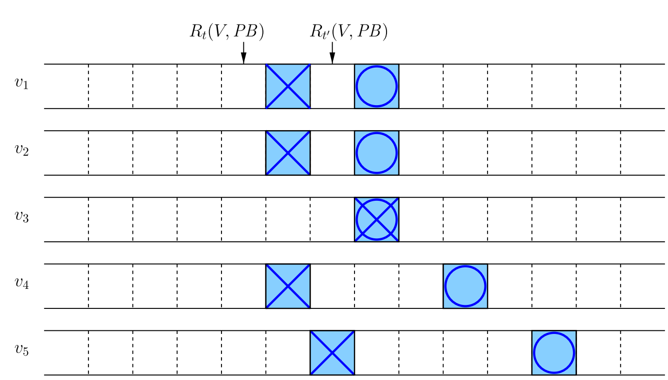

Fig. 3. Our proposed VoD scheme applied on the PB one in Fig. 2. It transmits the segments at the same times, but *only* if they are going to be "consumed".

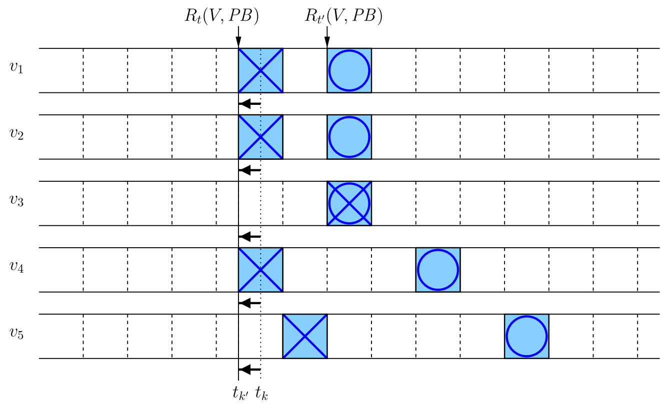

Fig. 4. Example of using the saved bandwidth (Proposition 4.1) for reducing waiting times. In this example, start time of time-slots is anticipated whenever possible (i.e. when there is no overlapping with previous transmissions) to adapt it to new requests. In this case, the slot starting at t_k is advanced to $t_{k'}$, and following slots are also advanced accordingly. Thus, when $R_t(V,\text{VoD})$ arrives, the start time of the next slot is set to that instant, so that transmissions to this request can start almost instantly (half a time-slot earlier than without the slot advancements).

Proof . By definition, our proposed VoD scheme schedules the transmissions that would be scheduled and consumed using the PB one, so that given a request of a video V at time t, $R_t(V,\text{VoD}) = R_t(V,\text{PB})$. Thus, the waiting time for any request is the same in both schemes, and the quality of service is also the same.

Regarding the bandwidth usage, by construction[1] $T'_i \subseteq T_i$ for all i at any time for any video V. This means that $S(V,\text{VoD})$ has equal or less scheduled segments than $S(V,\text{PB})$, i.e. the VoD scheme transmits equal or less segments than PB and therefore uses equal or less bandwidth. □

Note that using this transformation to obtain a VoD scheme does not prevent it from having lower waiting times than the PB scheme which it is based on. For instance, a possible way of using the saved bandwidth for transmitting segments earlier could be the one presented in Fig. 4. However, obtaining an automatic transformation to provide lower waiting times depends very much on the used PB scheme, since each one imposes its own restrictions on how to transmit segments.

5. Results

In this section, we show results based on simulations. First of all, and in order to provide experiments close to reality, we use a non-uniform request rate distribution along a day (Fig. 5). This distribution has been obtained from trials and it is used in previous works [18]. Notice that the request rate axis is referred to a single user making one request per week.

[1] All T'_i in $S(V,\text{VoD})$ are constructed by adding $ts \in T_i$ from $S(V,\text{PB})$, i.e. $T'_i = \emptyset \cup \{t_{R1} \in T_i\} \cup \{t_{R2} \in T_i\} \cup \ldots \cup \{t_{Rn} \in T_i\}$.

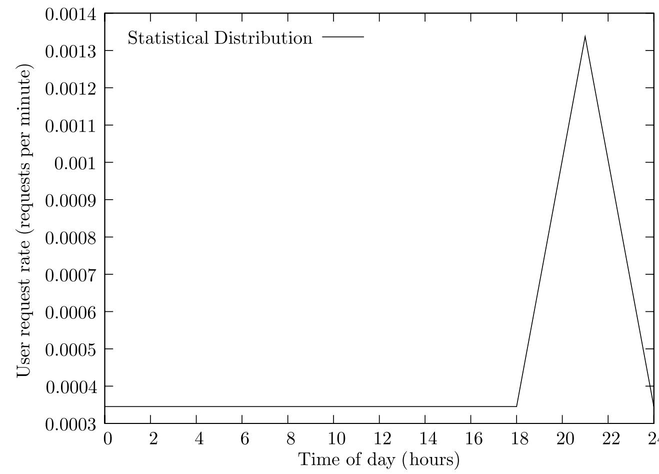

Fig. 5. Request rate distribution along a 24 h period.

For our experiments, we have used a system offering 500 videos to 4,000,000 users. These parameters provide, approximately, a 90 percentile of having one request for the least popular video each 90 min on the low load period (0–18 h). Note that this is a very high percentile, since it refers to the least popular video and popularity follows an exponential function. This means that more popular videos will be far more frequently requested. In order to compare our proposal with a good periodic broadcasting scheme, we have selected the *Polyharmonic Broadcasting* [15], which is optimal on bandwidth usage [9]. We have divided each video in 20 segments of 5 min each one. Fig. 6 shows bandwidth results applying our transformation on this method using these parameters for a complete day. Notice that we present our bandwidth results as percentages over the fixed bandwidth *Polyharmonic Broadcasting* would use. That is, the 100% value in this figure is the bandwidth *Polyharmonic Broadcasting* needs to transmit these 500 videos.

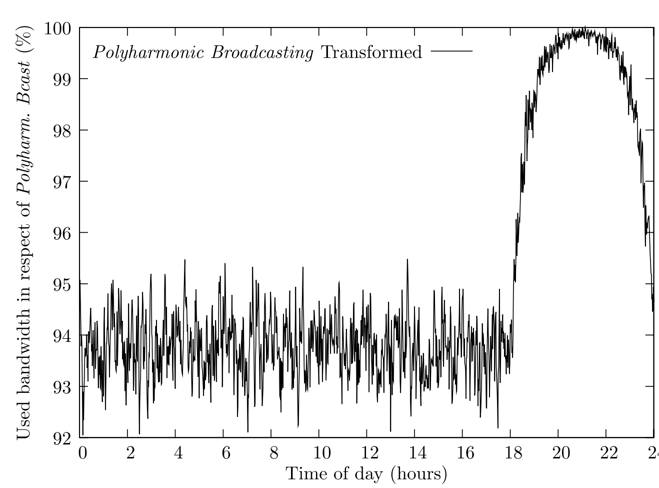

Fig. 6. Bandwidth usage of our transformed *Polyharmonic Broadcasting* in respect of the original one.

As proved above, bandwidth used by our proposal is equal or lower than *Polyharmonic Broadcasting* (100%). This lower bandwidth, even in respect of a bandwidth optimal method, is the first improvement our proposal provides. This freed bandwidth could be used for best-effort traffic, or could be effectively saved by making a previous estimation of the real required bandwidth. Differences in bandwidth along a day depend on the request rate (Fig. 5).

Since PB schemes broadcast media independently of requests, an accurate estimation of expected requests is required. That is, broadcasting for too few requests is a waste of bandwidth, and broadcasting for too much requests may be economically risked, since other business could offer more channels/videos and still be profitable. A major improvement when using our transformation is that the number of requests is far less important. In our case, if there are few requests the system uses less bandwidth. On the other hand, if there are many requests, the system performs at least as the PB one. This means we can relax the accuracy of estimations on expected requests.

Although not presented in this paper, modifying both videos and users in the same proportion affects the variance but do not vary mean results. Variance depends on the concrete requests at each given time, so that the more requests the less data variability.

About the waiting time, *Polyharmonic Broadcasting* requires all customers to wait exactly the same amount of time regardlessly of the timing of their request. Thus, by construction this scheme has this "limitation" and our proposed transformation inherits it too. This makes impossible, in this particular case, to improve waiting times.

6. Conclusions

As has been shown, given a periodic broadcasting scheme, it can be improved to a VoD one which will use equal or less bandwidth. Essentially, the transformed VoD makes an on-line reservation of the slots that will be used, depending on the incoming requests. We have also shown how the improved scheme can additionally offer lower serving times.

The obvious advantage is that the unused bandwidth can be used for example to provide best-effort services. However, there is a more important and perhaps less evident advantage: the number of clients required to obtain a cost-efficient system no longer has a direct dependency on the number of offered videos, i.e. Fig. 1 looses much of its relevancy. This opens the possibilities of offering video transmissions in new scenarios, e.g. a hotel offering a few movies to a few rooms, or a movie distribution company offering all its movies to a small geographical area.

Finally, further research on pricing for this way of scheduling is required. Since statistically we have lower bandwidth requirements, overbooking practices may be acceptable. This will require further trials. Additionally, some ways to dynamically modify the probability of requesting each video would also be very useful. For instance, one may lower the price of videos with already scheduled but not yet transmitted segments.

As a final conclusion, we state that current periodic broadcast schemes can be easily enhanced to have a lower bandwidth consumption and even offer lower waiting times. This improvement is based on transforming off-line scheduling algorithms to on-line ones, which can take profit of on-line information. This leads to a convergence of periodic broadcasting schemes and video-on-demand schemes.

References

[1] Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, A permutation-based pyramid broadcasting scheme for video-on-demand systems, in: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Hiroshima, Japan, 1996, pp. 118–126.

[2] Kevin C. Almeroth, Mostafa H. Ammar, On the use of multicast delivery to provide a scalable and interactive video-on-demand service, IEEE Journal of Selected Areas in Communications 14 (6) (1996) 1110–1122.

[3] Kevin C. Almeroth, Asit Dan, Dinkar Sitaram, William H. Tetzlaff, Long term resource allocation in video delivery systems, in: Proceedings of the IEEE Infocom, 1997, pp. 1333–1340.

[4] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker, Web caching and Zipf-like distributions: evidence and implications, in: Proceedings of the IEEE Infocom, March 1999.

[5] Ying Cai, Kien A. Hua, Khanh Vu, Optimizing patching performance, in: Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking, 1999, pp. 204–215.

[6] Steven W. Carter, Darrell D.E. Long, Improving bandwidth efficiency of video-on-demand servers, Computer Networks 31 (1–2) (1999) 99–111.

[7] Asit Dan, Dinkar Sitaram, A generalized interval caching policy for mixed interactive and long video workloads, Multimedia Computing and Networking (1996).

[8] Derek L. Eager, Mary K. Vernon, John Zahorjan, Minimizing bandwidth requirements for on-demand data delivery, Knowledge and Data Engineering 13 (5) (2001) 742–757.

[9] Lars Engebretsen, Madhu Sudan, Harmonic broadcasting is bandwidth-optimal assuming constant bit rate, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California, January 2002, pp. 431–432.

[10] L. Golubchic, J.C.S. Liu, R.R. Muntz, Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage servers, ACM Multimedia Systems Journal 4 (3) (1996) 140–155.

[11] Kien A. Hua, Simon Sheu, Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems, in: Proceedings of the ACM Special Interest Group on Data Communications (SIGCOMM), 1997, pp. 89–100.

[12] Li-Shen Juhn, Li-Ming Tseng, Harmonic broadcasting for video-on-demand service, IEEE Transactions on Broadcasting 43 (3) (1997).

[13] Li-Shen Juhn, Li-Ming Tseng, Fast data broadcasting and receiving scheme for popular video service, IEEE Transactions on Broadcasting 44 (1) (1998) 100–105.

[14] Jehan-Francois Pâris, Steven W. Carter, Darrell D.E. Long, Efficient broadcasting protocols for video on demand, in: Proceedings of the international symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, July 1998, pp. 127–132.

[15] Jehan-Frantois Pâris, Steven W. Carter, Darrell D.E. Long, A low bandwidth broadcasting protocol for video on demand, in: Proceedings of the International Conference on Computer Communications and Networks, October 1998, pp. 690–697.

[16] Juan Segarra, Vicent Cholvi, On-line advancements of transmission plans in video-on-demand, in: Proceedings of the International Conference on Distributed Computing Systems Workshops, Tokyo, Japan, March 2004, pp. 158–163.

[17] W. Tang, Y. Fu, L. Cherkasova, A. Vahdat, MediSyn: a synthetic streaming media service workload generator, in: Proceedings of the 13th International Workshop on Network and Operating System Support for Digital Audio and Video (ACM NOSSDAV), Monterey, CA, June 2003.
[18] Constantinos Vassilakis, Michael Paterakis, Peter Triantafillou, Video placement and configuration of distributed video servers on cable TV networks, Multimedia Systems 8 (2000) 92–104.
[19] Subram Viswanathan, Tomasz Imielinski, Metropolitan area video-on-demand service using pyramid broadcasting, ACM Multimedia Systems Journal 4 (4) (1996) 197–208.

Juan Segarra graduated in Computer Science at the Jaume I University (Spain) and received his doctorate in Computer Science at the same university in 2003. Also in this year, he joined the University of Zaragoza (Spain) where he is currently working. His research interests include QoS, media distribution and also cache memories in real-time systems and worst-cast execution times.

Vicent Cholvi graduated in Physics from the Universitat de Valéncia (Spain) and received his doctorate in Computer Science in 1994 from the Polytechnic University of Valencia (Spain). In 1995, he joined the Jaume I University in Castelló (Spain) where he is currently an Associate Professor. His interests are in distributed and communication systems.