Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 28 (2005) 1303–1316
computer communications
www.elsevier.com/locate/comcom

Adjustable broadcast protocol for large-scale near-video-on-demand systems

Jin B. Kwon[a,*], Heon Y. Yeom[b]

[a]Department of Computer Science, Sun Moon University, Kalsan 100, Tangjeong, Asan Chungnam 336-708, South Korea
[b]School of Computer Science and Engineering, Seoul Nat'l University, Sillim San 56-1, Kwanak, Seoul 151-742, South Korea

Received 16 April 2004; revised 7 January 2005; accepted 11 January 2005
Available online 25 January 2005

Abstract

Video-on-demand (VOD) is a service allowing customers to select video programs from a central server for viewing on a television or a computer screen. Traditional unicast VOD systems require huge amounts of the server network's bandwidth, so near-VOD systems using broadcast protocols have been proposed to reduce the bandwidth requirement. Although NVOD systems can service many users with less network bandwidth by broadcasting popular videos, the popularity of any given video is likely to vary widely as time goes on. However, the broadcast protocols cannot adapt to dynamic environments where the popularity distribution of videos changes, because of their fixed broadcasting schedule.

We introduce a dynamic broadcast protocol called *Adjustable Broadcast* (*AB*) based on on-line scheduling, which can adjust the number of channels being used on the fly. AB also exploits various *prefetching* strategies in the broadcast schedule to reduce the required bandwidth. And, we propose a dynamic channel allocation scheme using AB that minimizes the average viewer's waiting time. Using extensive simulation, we have demonstrated the effectiveness of AB and the adaptability of our dynamic channel allocation.

Keywords: Multimedia systems; Multimedia communication; Video broadcast; Scheduling

1. Introduction

Video-on-demand (VOD) is a service allowing customers to select programs from a central server for viewing on a television or a computer screen. VOD can be used for entertainment (ordering movies transmitted digitally), education (distance learning) and other purposes, although it is not yet widely implemented. In traditional unicast VOD systems, each customer is served by an individually allocated channel. Although these systems can respond to requests immediately, the server network bandwidth is rapidly depleted. This is the reason why these VOD systems are more expensive than the rivals such as pay-per-view and video-cassette rentals.

This situation has resulted in much work aimed at reducing the bandwidth requirements of VOD services, by using multicast or broadcast networks. In this research area, it is assumed that clients can receive streaming data from several channels simultaneously. The proposed protocols fit into one of two categories, *reactive* and *proactive* approaches. Reactive protocols assume that the video server processes individual customer requests without anticipating the number of requests for a video [2,5,6,7,10,11]. Whenever several users request the same video in close succession, the server tries to transmit all data to them by using only a shared stream. On the other hand, proactive protocols only deal with popular videos, and transmit the various segments of each video according to a deterministic schedule. Thus, their bandwidth requirements are not affected by the request arrival rate for a given video. Hence the proactive protocols result in high performance in heavy demand. These protocols are also called *broadcast*

* Corresponding author. Tel.: +82 41 530 2258; fax: +82 41 530 2876.
E-mail addresses: jbkwon@sunmoon.ac.kr (J.B. Kwon), yeom@snu.ac.kr (H.Y. Yeom).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.01.004

protocols [1,9,12,13,15,18]. In Bradshaw et al. [3] reported on the implementation, measurement, and analysis of a working video server testbed implementing both proactive approaches and reactive approaches.

The problem with both protocols is that the frequency of requests for any given video is likely to vary widely as time goes on. While reactive protocols will perform very well for videos in low demand, they run the risk of overloading the server when a video is in high demand. On the other hand, broadcast protocols will perform very well with high-demand videos, but will use unnecessarily channels in all other cases. To overcome this problem of the proactive approach, some protocols to combine the reactive and the proactive approaches have been proposed. These protocols include *dynamic skyscraper broadcast* [8], *universal distribution* (UD) [16], *dynamic heuristic broadcast* (DHB) [4], and *channel-based heuristic distribution* (CBHD) [20], *recursive frequency-splitting* [17]. Yang et al. proposed a *borrow-and-return* model that shares idle channels to reduce the viewer's waiting time [19].

These dynamic broadcast protocols reduce the bandwidth wasted in low demand by dynamically scheduling data segments for requests arriving within a specified time interval, i.e. a *time slot*. Therefore, if there is no request within the current time slot, no new data segment is scheduled and there is no unnecessary usage of channels. However, they have three problems. First, they rely on the existence of requests during a slot to determine whether a video is popular or not. Assume that the request rate for a video V_i is 20 requests/min. and that of another video V_j is 0.5 requests/min. Then, when the time slot is 5 min, we expect that there will be one or more requests in a time slot for both videos. The existing dynamic protocols handle the two videos, V_i and V_j, without considering the difference in the request rates. That is the reason why the dynamic protocols cannot adapt well to environments in which the popularity of videos is dynamically changing. Second, they do not show how to manage the network bandwidth for servicing multiple videos. In other words, because they are not able to control the number of channels allocated to each video, they have no way to use the saved channels from less popular videos to service more popular videos. Third, their performance at high request arrival rates is not as good as that of the best proactive protocols.

A different approach for broadcast VOD systems is required to cope with dynamic environments. In fact, it is reasonable to regard the popularity of a video as being relative to other videos. In other words, that a video is in heavy demand does not mean that the request rate for the video is above a specific value. It means that the demand for that video is higher than those of other videos. Given a limited network bandwidth, we should develop an optimal channel allocation for each video based on the popularity distribution. In the broadcast VOD systems, allocating more channels to a video means to reduce users' waiting time for watching the video. Since the popularity distribution of videos is hard to be known in advance and even changes as time goes by, it is desirable to allocate channels on the fly to adapt to the distribution shift. In addition, this dynamic allocation is also beneficial and required in that channel allocation can be controlled manually or automatically based on a policy other than the popularity without interrupting the service. However, the dynamic channel allocation is impossible in the existing broadcast protocols due to their inflexibility. That is why the issues on dynamic channel allocation have not been tackled. To make this possible, the number of channels allocated to a video should be dynamically adjustable at run time according to changes in popularity distribution. That is, the channel distribution among the videos should dynamically adapt to the popularity distribution. In this paper, we propose a video broadcast protocol, called *adjustable broadcast* (AB) that can dynamically adjust the number of channels allocated to each video at run time. To implement AB, we devised a *flexible scheduling* mechanism to enable this adaptability, and various *prefetching* strategies in the broadcast schedule to enhance the performance. To show how AB can be applied to practical applications, we also introduce a basic channel management mechanism to decide how many channels should be allocated to each video based on its popularity. All the proposed protocols have been verified using extensive simulation.

The rest of this paper is organized as follows: we discuss related work and our approach in Section 2. In Section 3, we introduce the on-line scheduling, and in Section 4, we present flexible scheduling and prefetching mechanisms applied to our broadcast protocol, and devise a channel management scheme using the proposed broadcast protocol. Section 5 demonstrates our simulation results, and finally we conclude our work in Section 6.

2. Broadcast protocols

A client must wait for the next broadcast to start, to watch a video in a broadcast NVOD system, which broadcasts each video periodically. The broadcast interval is called the *broadcast cycle*, and the time between the request arrival and the playback start is defined as the *service latency*. Hence, the worst-case service latency is the broadcast cycle. Once a user begins to watch a video, the video must be continuously played. Assume that the whole video consists of d-minute segments, and each data segment is transmitted at the same rate as the playback rate b (Mbps) of the video. That is, the time to transmit a segment is d minutes. The client is also assumed to be able to play the segment while receiving it. Most of the existing video broadcast protocols have made efforts to reduce the bandwidth required to ensure the given worst-case service latency and continuous playback.