A P2P Broadcasting Scheme for Video on Demand Services

Hwang Kyu Choi, Yi Qi Gui, Ey Suk Jung,
Chi Hoon Lee, In Bum Jung, and Chang Yeol Choi

Dept. of Computer Science and Engineering, Kangwon National University
Chuncheon, Kangwon-Do, Korea 200-701

Abstract *In this paper, we propose a new broadcasting scheme, called P2P broadcasting, for eliminating the service latency time of the existing broadcasting schemes. Our broadcasting scheme is based on the simplest staggered broadcasting protocol and then we use peer to peer caching technique, in which a number of clients that share a segment of a stream delivered from the same channel forms a group. Each client in a group plays a role of a prefix caching server to deliver the first segment of the stream for other clients. During the fixed time interval of one segment, each client in a group caches a portion of the first segment received from another channel that starts to transmit the first segment of the stream, and then they support the first segment to new clients during the time interval of the segment. As a result of the P2P caching, our broadcasting scheme can eliminate the service latency of the broadcasting scheme along with less storage requirement than the existing broadcasting schemes.*

Keywords : Caching, Peer to peer, Periodic broadcasting, Proxy, Video on demand

1 Introduction

Video-on-Demand(VOD) service is one of the most important technologies for many multimedia applications, such as home entertainment, digital video libraries, distance learning, electronic commerce, and so on. A typical VOD service allows that a number of remote clients playback a desired video from a large collection of videos stored in one or more video servers. The main bottleneck for a VOD service is the storage bandwidth of the VOD server and the network bandwidth connecting to the VOD server to the client due to the high bandwidth requirements and long-lived nature of digital video.

Previous research has shown that performance of VOD system can be greatly improved through the use of multi-casting or broadcasting scheme[1]. Multi-casting schemes try to let user share the same stream of data as much as possible[8,9]. While some of the multi-casting approaches can provide immediate service and save server bandwidth by avoiding unnecessary transmission of data, they are subject to data loss and can not guarantee on time delivery of data if user requests are bursty or too high. Broadcasting schemes can address this problem by periodically transmitting video segment and guarantee service latency within certain amount of time[2-7]. The idea behind periodic broadcasting schemes is to divide the video into a series of segments and broadcast each segment periodically on dedicated server channels. While user is playing the current video segment, it is guaranteed that the next segment is downloaded on time and the whole video can be played out continuously. User will have to wait for the occurrence of the first segment before they can start playing the video. User waiting time is usually the length of the first segment.

Staggered broadcasting[12] is the simplest broadcasting protocol proposed in the early days. It allocates K server channels each with bandwidth b to transmit the whole video. The beginnings of each video replica are staggered evenly across the

This paper was supported by grant No. B1220-0401-0203 from the University Fundamental Research Program of the Ministry of Information & Communication in Republic of Korea.

channels at fixed time intervals. Client service latency is *S/K*, where *S* denotes the total length of the video, and it could not be improved without the expense of linear increase in the corresponding server bandwidth. Some more efficient broadcasting protocols, such as *Skyscraper Broadcasting*[3], have been proposed[2-7]. All these protocols divide each video into *n* segments that are simultaneously broadcast on different channels. One of these channels transmits nothing but the first segment of the video. The other channels transmit the remaining segments at their designated bandwidth. When clients want to watch a video, they wait for the beginning of the first segment from the first stream. While clients start watching that segment, they also start downloading enough data from the other stream so that it will be able to play each segment of the video in turn.

In the case of popular video, they can reduce the server bandwidth significantly, because the broadcasting scheme transmits a stream to many users simultaneously without additional bandwidth requirement. However, all the broadcasting schemes have a common problem to achieve the true VOD service due to their service latency times. To address the drawback, many broadcasting schemes[2-7] have been designed to keep the smaller service delay by making the first segment as small as possible, so that the broadcasting protocol becomes complicated and requires a large amount of client storage space.

In this paper, we propose a new broadcasting scheme, called *P2P broadcasting*, for eliminating the service latency time of the existing broadcasting schemes. Our broadcasting scheme is based on the simplest staggered broadcasting protocol and then we use peer to peer (P2P) caching technique in order to realize the true VOD service. In the P2P caching, a number of clients that share a segment of a stream delivered from the same channel forms a group. Each client in a group plays a role of a prefix caching server to deliver the first segment of the stream for other clients. During the fixed time interval of one segment, each client in a group caches a portion of the first segment received from another channel that starts to transmit the first segment of the stream, and then they support the first segment to new clients during the time interval of the segment. As a result of the P2P caching, our scheme can eliminate the service latency of the broadcasting scheme along with less storage requirement than the existing broadcasting schemes.

The remainder of this paper is organized as follows. Section 2 describes the related work to the broadcasting schemes for VOD systems. In section 3, we represent our P2P broadcasting scheme. Section 4 evaluates the performance of the proposed scheme. Finally, we conclude this paper in section 5.

2 Related Works

During the past decade, there have been a large amount of research works on the topic of VOD systems. We first investigate the related research works for VOD systems, especially focused on the techniques for reducing server bandwidth requirement. Although a VOD server has high capacity of computation and storage resources, the performance of VOD system can be constrained by increasing more clients. In order to overcome the drawbacks caused by limited server bandwidth, many techniques based on broadcasting[2-7] or multi-casting scheme have been proposed[8,9].

Staggered Broadcasting[12] is the simplest broadcasting protocol proposed in the early days. It allocates *K* server channels each with bandwidth *b* to transmit the whole video. The beginnings of each video replica are staggered evenly across the channels. Client access latency is *S/K*, where *S* denotes the total length of the video, and it could not be improved without the expense of linear increase in the corresponding server bandwidth.

Some more efficient broadcasting protocols have been proposed. All these protocols share a similar organization. They divide each video into *n* segments that are simultaneously broadcast on different channels. One of these channels transmits nothing but the first segment of the video. The other channels transmit the remaining segments at their designated bandwidth. When clients want to watch a video, they wait for the beginning of the first segment from the first stream. While clients start watching that segment, they also start downloading enough data from the other stream(s) so that it will be able to play each segment of the video in turn.

In *Pyramid Broadcasting*(PB) protocol[2], the segment size of the videos follows a geometrical series and different videos are mingled together in each logical channel. To provide on time delivery of the videos, each segment channel has to transmit the segments in a very high rate and client I/O bandwidth and storage requirement are also high. Clients can download the next segment at its earliest occurrence and at any time they download at most two consecutive channels. To address the problem of high client side requirement in PB, *Permutation-based Pyramid Broadcasting*(PPB) [4] was proposed. Instead of transmitting a segment in a very high bandwidth, PPB multiplexes the segment channel into *P* subchannels and transmitted them in P times lower rate. The *P* substreams are staggered with each other to meet the same timing requirement as in PB. Another important broadcasting protocol is the *Skyscraper Broadcasting*(SB) protocol[3]. SB transmits each segment in the video consumption rate *b* and its segment series progression is much lower but still meets the timing requirement. Client needs to download from at most two streams at any time. The client disk storage requirement is constrained by the size of the last segment.

3 P2P Broadcasting Scheme

In this section, we first present the basic staggered broadcasting scheme and then propose a new *P2P broadcasting* scheme that jointly uses the staggered broadcasting and a peer to peer caching approach for eliminating the service latency of the existing broadcasting schemes.

3.1 Staggered Broadcasting

In our broadcasting scheme, we basically use the staggered broadcasting scheme[12] to transmit a regular stream from a server to clients. As shown in Figure 1, the staggered broadcasting starts video broadcasts over the allocated channels at fixed time intervals. Assume that there are *K* channels dedicated to a video of length *S*. The *K* channels are labeled as *0, 1, 2, …, K-1*. The video broadcast starts every *S/K*(= *d*) seconds and each channel broadcasts the whole video repeatedly. If the broadcast of a video starts at channel *0* at the system setup time t_0, the broadcast starts over channel *k* at $t_0 + kd$ and is repeated. Therefore, the video is broadcast every *d* seconds over *K* channels and accordingly, the maximum service latency is *d* seconds. Here, the *K* video chunks of length *d* that make up the video are called *segments*. During a segment time (i.e., *d* seconds), all *K* segments are broadcast over different channels.

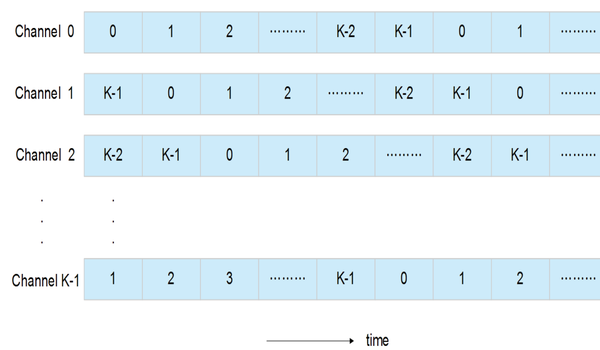

Figure 1. Staggered broadcasting scheme

Since the staggered broadcasting receives a segment from one channel at a time, it is not necessary additional client storage, but on the other hand most of the existing broadcasting techniques[2-7] require additional storage space and channel bandwidth to receive multiple segments from different channels simultaneously. Although the staggered broadcasting scheme is very simple and efficient for both the storage and the bandwidth requirements, it has a long latency time, so that it is difficult to achieve a true or near VOD service.

3.2 Overview of P2P Broadcasting

In this paper, we propose a new *P2P broadcasting* scheme based on the simple staggered broadcasting protocol to eliminate the service latency time along with minimizing the client storage requirement by using P2P caching scheme[10,11]. Figure 2 depicts the overall system architecture for our P2P broadcasting scheme, where the VOD system consists of a central VOD server and P2P client group. Each client has its own storage for caching the segments of a stream from P2P client.

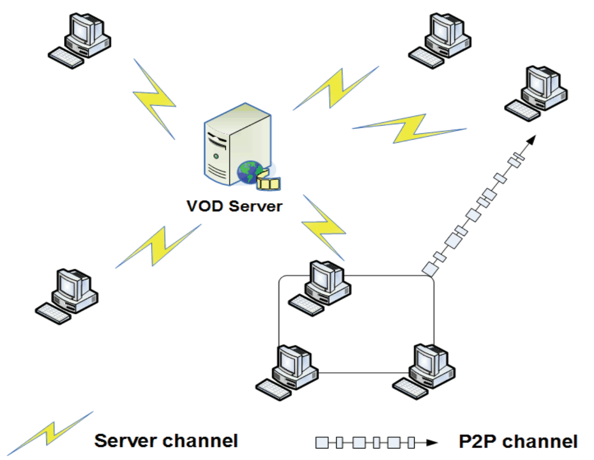

Figure 2. System architecture of P2P broadcasting

In the P2P caching, a number of clients that share a segment of a stream delivered from the same channel forms a group. Each client in a group plays a role of a prefix caching server to deliver the first segment of the stream for other clients through the P2P channel. While receiving the regular stream from the server through broadcast channel, each client in a group caches a portion of the first segment received from another channel that starts to transmit the first segment of the stream. Since all K segments are broadcast over different channels during a segment time d, each client can receive the first segment from one of the K channels at any beginning of segments. During the segment time d, these clients support the first segment to new clients that request the same video during the same segment time interval. After that, the client releases the cached first segment from the storage. As a result of the P2P caching, our broadcasting scheme can eliminate the service latency of the staggered broadcasting scheme along with minimizing client storage requirement.

3.3 Service Scenario of P2P Broadcasting

Figure 3 describes the basic service scenario of P2P broadcasting scheme, where C_i denotes *i*th client starting at time t_i and group i means a set of clients that share a segment received from the same channel. All the clients in a group play two roles of both the playing client and the prefix server for other clients. As soon as a client request arrives, the client is assigned to a broadcast channel that currently transmits the first segment for receiving the broadcast stream from the server. The client starts to play the stream by receiving the first segment from clients in another group and simultaneously receiving the broadcast stream from the server. While the client plays the prefix stream, the regular stream from the server should be saved in a buffer to play it after playing all the prefix stream. In Figure 3, *PB* denotes the playing buffer for storing the regular stream and playing it sequentially. For each client C_i, the required amount of the buffer $PB_i = t_i - t_s$ in time.

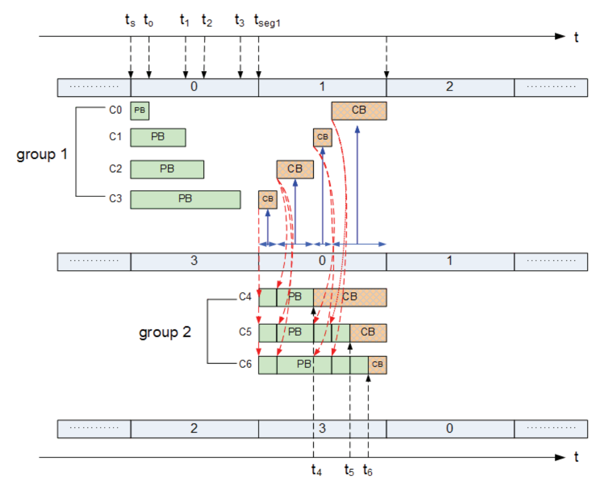

Figure 3. Service scenario of P2P broadcasting

As a prefix server, each client in a group caches a portion of the first segment received from another channel that starts to transmit the first segment of the stream and then they support the first segment to new clients during the time interval of the segment. We assume that all the clients in a group have the same amount of buffer storage as same as one segment size. The whole client buffer is divide by two partitions, one is the playing buffer and the other is the caching buffer for prefix caching received from other clients. The caching buffer is distributed into all the clients in a group, where the rest of the playing buffer in the whole buffer is used for the caching buffer. In Figure 3, *CB* denotes the caching buffer and for each client C_i the required amount of the buffer $CB_i = t_{seg} - t_i$ in time. The total amount of the caching buffer in a group is the same as the one segment size.

3.4 Failure Recovery

In P2P broadcasting scheme, the departure of a client or an underlying network failure, such as link breakdown or available bandwidth fluctuation, can disrupt the delivery of the first segment from a client to other clients. The proposed scheme provides failure recovery by replacing the failure client to another client in the same group or the central VOD server. When a client fails in our P2P broadcasting scheme, the next client of the failed client has to be connected to the forward client of the failed client in order to receive the prefix seamlessly. For reforming the chain of the clients without the failed client, each client should store additional streams in its available buffer until two consecutive requests arrive. Figure 4 describes such a situation for leaving C_2 from the group *1*. As soon as the client C_2 leaves C_1 starts to cache the prefix and then deliver the prefix instead of C_2.

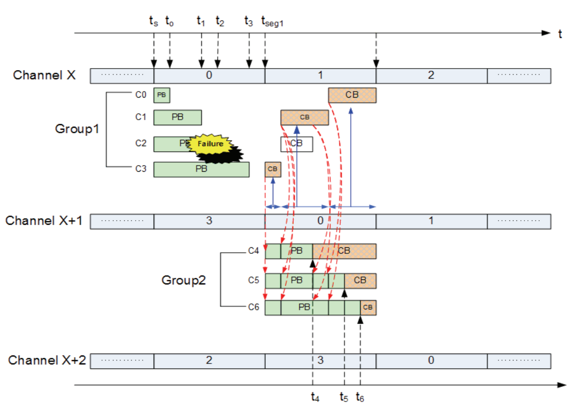

Figure 4. Failure recovery procedure of P2P broadcasting scheme

4 Performance Evaluation

In this section, we evaluate the performance of the P2P broadcasting scheme through simulation experiments. We examine the client buffer requirement varying the number of channels and the request interval time. We then compare the performance of our scheme with the existing broadcasting scheme with respect to the client buffer requirement and service latency time. The parameter used in the simulation study is shown in Table 1. In our simulation, we suppose that the client request rate to VOD server follows the Poisson distribution with the default request frequency λ and there is no failure client. We also suppose that access pattern to videos follows Zipf-like distribution with skew factor θ and the default skew factor for our simulation is 0.271. A uniform distribution corresponds to the value of 1, and a value of 0 represents a highly skewed distribution. The request probability to video *i* obtains by the following formula, $P_i = C / i^{1-\theta}$.

Table 1. Performance parameters

Parameter	Default	Variation
Number of videos	1	N/A
Mean inter-arrival time (second)	5	5~50
Video length (minute)	60	N/A
Number of broadcast channels	6	6~10
Skew factor	0.271	N/A

Figure 5 presents the effect of mean request interval times for the client buffer requirement. We fix the video length at 60 minutes and the number of broadcast channels is varied from 6 to 9. The graph shows that client buffer requirement increases as the client request interval time increases. It is because the amount of the prefix stream to be cached increases as the request interval time is longer. The client buffer requirement decreases as the number of the broadcast channels increase. According to increase of the number of the broadcast channels, the size of the first segment is getting smaller, therefore, the amount of the cached prefix stream are getting smaller also.

Among the existing broadcasting schemes, Skyscraper Broadcasting has shown the best performance with respect to the requirement of client buffer. We compare the buffer requirement of our broadcasting scheme with that of Skyscraper Broadcasting. In Skyscraper the buffer requirement and the service latency time are derived as follows[3] :

Service latency time D_1

$$= \frac{D}{\sum_{i=1}^{k} \min((f(i), W))} \text{ seconds} \qquad (1)$$

Maximum buffer requirement

$$= D_1 \times (W-1) \text{ seconds} \qquad (2)$$

In the above formulas, D denotes the video length, and W is the width of the skyscraper which can be controlled to achieve the desired service latency time. A smaller W corresponds to a larger D_1 and also to decrease the buffer requirement.

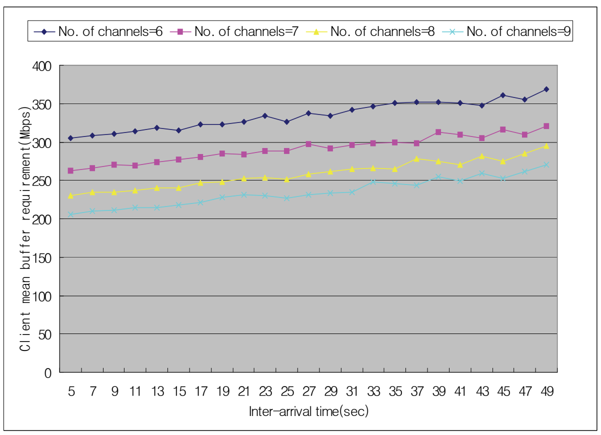

Figure 5. Client buffer requirement for varying the mean request interval

Figure 6 represents the effect of the number of channels for the buffer requirement. We fix the video length at 60 minutes and the number of broadcast channels is varied from 5 to 65. In Figure 6, the number of channels increases, the size of the first segment decreases in both schemes. According to increase the number of broadcast channels, therefore, the buffer requirement is smaller.ents for varying the number of channels

We compare the performance of our scheme about service latency time by change of number of channel with the Skyscraper Broadcast scheme. Figure 7 represents the effect of number of channel for the service latency time. The graph shows that P2P broadcasting scheme has zero service latency time regardless of the number of channels. The service latency time of Skyscraper Broadcasting scheme decreases as the number of channels increases, because the size of D_1 decreases as the number of broadcast channels increases.

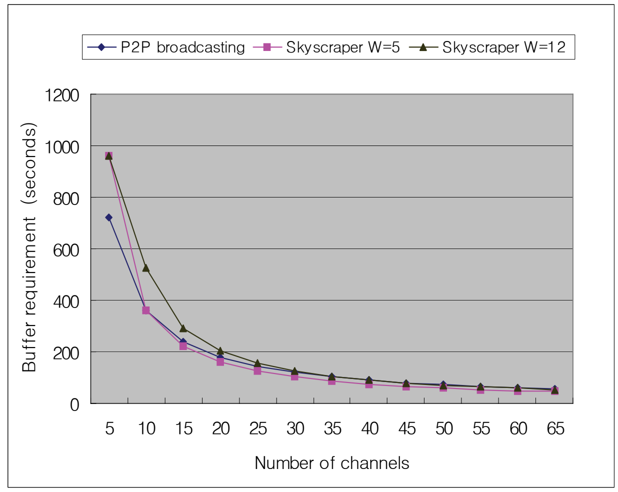

Figure 6. Comparison of the buffer requirements for varying the number of channels

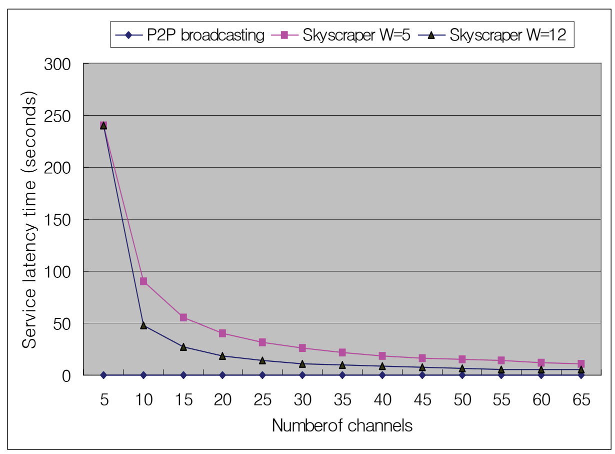

Figure 7. Comparison of the service latency times for varying the number of channels

5 Conclusion

The main bottleneck for a VOD service is the storage bandwidth of the VOD server and the network bandwidth connecting to the VOD server to clients. Periodic broadcasting is one of the most efficient techniques to overcome the bottleneck of the VOD system, because the broadcasting scheme transmits a stream to many users simultaneously without additional bandwidth requirement. However, all the broadcasting schemes have a common problem to achieve the true VOD service due to their service latency times. In this paper, we proposed P2P broadcasting for eliminating the service latency time of the existing broadcasting schemes. Our broadcasting scheme is based on the simplest staggered broadcasting protocol and then we used the P2P caching technique in order to realize the true VOD service. In the P2P caching, a number of clients that share a segment of a stream delivered from the same channel forms a group. Each client in a group plays a role of a prefix caching server to deliver the first segment of the stream for other clients. During the fixed time interval of one segment, each client in a group caches a portion of the first segment received from another channel that starts to transmit the first segment of the stream, and then they support the first segment to new clients during the time interval of the segment. As a result of the P2P caching, our broadcasting scheme can eliminate the service latency of the broadcasting scheme along with less storage requirement than the existing broadcasting schemes.

References

[1] A. Hu, "Video-on-Demand Broadcasting Protocols: A Comprehensive Study," Proc. of IEEE INFOCOM, April 2001.

[2] S. Viswanathan and T. Imielinski, "Pyramid Broadcasting for Video on Demand Service," Proc. of ST/SPIE Conf. on Multimedia Computing and Networking, 1995.

[3] K. Hua and S. Sheu, "Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan VoD Systems," ACM SIGCOMM, 1997.

[4] C.C Aggarwal, J. L. Wolf, and P. S. Yu, "A Permutation-based Pyramid Broadcasting Scheme for Video-on-Demand Systems," Proc. of the IEEE International Conf. on Multimedia Systems'96, Hiroshima, Japan, June 1996.

[5] J. F. Paris, S. W. Carter, and D. D. E. Long, "A Hybrid Broadcasting Protocol for Video on Demand," Proc. SPIE Conference on Multimedia Computing and Networking, San Jose, USA, Jan. 2000.

[6] L. Juhn and L. Tseng, "Harmonic Broadcasting for Video-on-Demand Service," IEEE Transactions on Broadcasting, Vol. 43, No. 3, 1997.

[7] L. Juhn and L. Tseng, "Fast Data Broadcasting and Receiving Scheme for Popular Video Service," IEEE Transactions on Broadcasting, Vol. 44, No. 1, 1998.

[8] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services," Proc. of ACM Multimedia, Bristol, U.K. Sep. 1998.

[9] C. J. Kwon, C. K. Choi, and H. K. Choi, "An Improved Patching Scheme for Video-On-Demand Servers," Proc. of the International Conf. on Parallel and Distributed Processing Techniques and Applications, June 2004.

[10] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation, IEEE ICME, 2003.

[11] S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix Caching for Multimedia Streams," Proc. of IEEE INFOCOM, March 1999.

[12] J. B. Kwon and H, Y. Yeom, "Providing VCR Functionality in Staggered Video Broadcasting," IEEE Transaction on Consumer Electronics, Vol. 48, No. 1, 2002.