Provision of VCR Functionality for Video-on-Demand System with Cooperative Clients in Broadcast Environment

K.M. Ho, K.T. Lo

***Abstract*—In this paper, we investigate a possible solution for supporting VCR functionality in broadcast environment with peer-to-peer paradigm. Unlike the conventional approaches that the merging process is done by the central server during VCR actions, it is handled in P2P manner in our framework. The main focus of this paper is to study how system parameters, such as arrival rate, batching time, departure rate, client upload bandwidth and the population of free-rider affect the performance of the system.**

***Index Terms*—Video Broadcast, VCR Functionality, Video-on-Demand (VoD), Peer-to-Peer (P2P)**

I. Introduction

With the explosive growth of the Internet, the demand for Video-on-Demand (VoD) applications has been rapidly increasing in recent years. With VoD, clients can enjoy their favorite videos at arbitrary times via public networks. Such systems are required to store hundreds of videos and serve thousands of clients simultaneously. In order to provide cost effective and scalable solutions for such systems, a number of transmission strategies such as peer-to-peer (P2P) and broadcast/multicast have been proposed in the past decade.

In P2P architecture [1], each endpoint called peer is operated as client and server simultaneously. Each new incoming peer is required to make a partnership with other peers in the system in such a way that a P2P network is constructed. Then, each peer can retrieve what it wants from the system and forward what it has to the system over this network. As the successful deployment of IP multicast/broadcast delivery [2], people have also exploited the broadcast capability of a network to support VoD services. In broadcast VoD [1], a video is first partitioned into a number of segments. Then, each segment is transmitted over a dedicated broadcast channel periodically. Under the predefined schedule, clients can fetch the desirable segments from the broadcasting channels. Different from the broadcasting system, in the multicast VoD, customers arriving closely enough are grouped together and served by a single multicasting channel [1]. Recently, a number of transmission protocols such as Peer-to-Peer Batching (PPB) policy [3] have been proposed to enhance the system performance by coupling broadcast/multicast transmission strategy with P2P paradigm.

Although the broadcast/multicast approach can use the system resources more efficient than P2P, it only provides near VoD (NVoD) services to clients. In order to support true VoD (TVoD) services, buffer management schemes [4-5] and merging schemes [6] were proposed to provide interactive function to clients. Obviously, these schemes consume extra central server resources. To tackle this shortcoming, it introduces an interesting question if hybrid approach can bring benefits to the system performance. In this paper, we have investigated the solution for supporting VCR functionality in broadcast environments with P2P. Unlike the conventional approaches [4, 6] that the merging process is achieved by the central server, VCR actions are handled in P2P manner in our framework. The main focus of this paper is to study how system parameters, such as arrival rate, batching time, departure rate, client upload bandwidth and the population of free-rider affect the performance of the system. The rest of this paper is organized as follows: Section 2 reviews the current protocols on supporting VCR functions in broadcast environments. In Section 3, we will describe the details of the proposed policy that supports VCR functions by P2P paradigm. Performance evaluation of the proposed schemes by computer simulation will be discussed in Section 4. Finally, some concluding remarks are given in Section 5.

II. Related Works

In order to support interactive functions, such as Pause (PAU), Jump Forward/Backward (JF/JB) and Fast Forward/Backward (FF/FB), in broadcast environment, a number of novel schemes have been proposed recently [4-7]. As described in [4-7], the system can fully support the PAU function with client buffer with the size of one time slot (i.e. batching time). During the PAU operation, the playout of video frames are suspended but the client buffer continues to cache the video data from the current broadcast group. When the buffer fills up, the client will switch to another broadcast group. However, it is still not satisfactory to achieve the smooth VCR action by using such a technique. Generally, the client has to jump to the nearest eligible point with respect to the specified destination if the required contents cannot be found in the buffer, or has to switch to other broadcast groups after resuming from the interactive actions [4]. In order to provide a smooth VCR operation, the merging algorithm has

Manuscript received December 08, 2011; revised January 03, 2012.

K.M. Ho is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: enkmho@hotmail.com).

K.T. Lo is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: enktlo@inet.polyu.edu.hk).

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

been proposed [4, 6]. The idea is to open the contingency channels to support customers who cannot merge into broadcast groups. Then, client downloads will be transferred across to the contingency channel. At the same time, the current video contents are being received from the broadcast channel. Once the play point is in the buffer, the contingency channel is released. In addition to use the contingency channel, a novel management scheme named Active Buffer Management (ABM) [5] was developed to support VCR function. With this approach, a client will obtain video data from several broadcasting channels simultaneously to maintain the play point in the middle of the buffer. To enhance the performance of ABM, Greedy Channel Management (GCM) scheme has been defined in [6]. This scheme combines both the ABM and the contingency channel to take advantage of using multiple loaders and minimize the requirement of contingency channel.

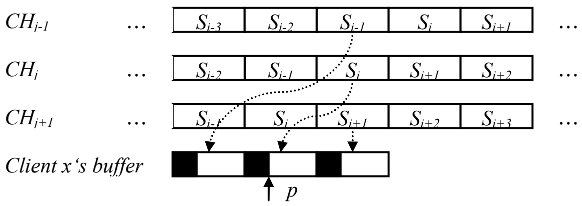


Fig.1 Buffer scheme of the proposed system

III. Proposed VRC Mechanism

The VCR mechanism in the proposed scheme is similar to the aforementioned approaches described in Section II. Client buffer is used to support the PAU action and limited JF/JB and FF/FB actions. The contingency channel is allocated whenever the merging operation is needed after the completion of VCR operations. However, unlike the previous schemes that the contingency channel has to be assigned by the central server, it is handled in P2P manner in our framework.

Assume that video *i* is *L* seconds long and it is transmitted in a staggered manner. If the phase delay (or batching time) between the two video channels is d seconds, the system is required to allocate $N=L/d$ number of broadcast channels. For simplicity of the following discussion, it is assumed that the video is partitioned into a number of segments, each of which has the size of *d* seconds. Each broadcast channel broadcasts each segment of the video in sequence. Therefore, as shown in Fig. 1, when broadcast channel *j* (CH_j) is broadcasting segment *i* (S_i), CH_{j-1} and CH_{j+1} are delivering $S_{(i-1)\%N}$ and $S_{(i+1)\%N}$ respectively. In the proposed policy, the buffering scheme follows ABM that the buffer is divided into three parts. Each part of the buffer, in turn, holds the past, the current and the future segments of the video. During the normal playback (i.e. no VCR action), each client fetches three segments from broadcast channels as the ABM scheme. As shown in Fig. 1, client *x* is required to get $S_{(i-1)\%N}$, S_i and $S_{(i+1)\%N}$ simultaneously if its current playback point (*p*) is in S_i. Additionally, each client needs to execute the Gossip mechanism [8] to make partnership with other clients fetching the same segments. Also, the system employs some location-aware strategies such as the LTM technique [9] for partnership optimization to seek the latest partner and eliminate slow connections. Thus, a well-organized overlay network for this segment is formed. Generally, each client is required to join three P2P networks and to leave these networks when the client no longer needs to use the service from the corresponding P2P network.

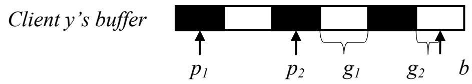


Fig.2 JF/JB Buffering Operation

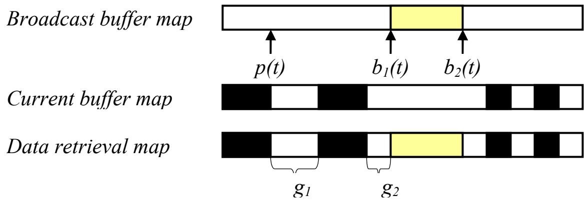


Fig.3 FF/FB Buffering Operation

Now we consider the VCR actions. In the proposed policy, PAU action can be accomplished as usual by ABM buffering technique. Handling of JF/JB actions is similar to the GCM approach. After resuming from these two actions, the client obtains the segments from the broadcast channels as GCM. But, it needs to reserve one loader for patching in P2P manner. The client finds out gaps in the buffer between the nearest channel position and the resuming point. As shown in Fig 2, client y issues a JF command to move the playback point from *p1* to *p2*. It can be seen that there are two gaps, *g1* and *g2*, between the nearest channel position *b* and the resuming point *p2*. Then, this client requests for the video data within the gap in order by flooding a request message, including the information of the start and end point of each gap and the playback time, over the corresponding P2P network for that segment. However, in case of the request failing, client y has to generate another request to the central server to complete the patching operation directly as GCM. Now, we are going to discuss how to handle of FF actions as illustrated in Fig. 3 (the operation of FB is the same but in reverse direction). Assume that the playback pointer of the client is at *p(t)* and the nearest channel position is at $b_1(t)$ when he/she issues a FF action. It first calculates the time when the playback pointer and the download point of the nearest channel are the same. In this example, it is $b_2(t)$. Therefore, there are $b_2(t)$- $b_1(t)$ units of data that can be fetched from the broadcast channel during the FF action and we have a broadcast buffer map as shown in the figure. Then, the client performs logic OR operation to mix the broadcast buffer map and the current buffer map to form a data retrieval map. As a result, there are a number of buffer gaps appeared (i.e. g_1, g_2 and g_3) and the client requests for the video data within the gap over the corresponding P2P network. When the playback pointer reaches $b_2(t)$, this operation is repeated and so on until the FF action completes. Similarly, if the request cannot be satisfied by the P2P network, the client has to generate another request to the central server to obtain the required data.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Preliminary Simulation Results

A simulation model is built to evaluate the performance of the interactive VoD system. Since the requests for different videos are independent, we simply consider a single video case. It is assumed that the length of the video is two hours. Unless otherwise specified, the number of broadcast channels (N) in the model is 20 and thus the client buffer is required to store 18 minutes worth of video data. The arrival pattern and the departure pattern of the system are modeled as the Poisson Process with rate of λ and the exponential distribution with rate of μ respectively. Each client has outbound bandwidth of B_{up}. In order to model the user's activity, it is assumed that the operation is in one of two states: normal and interactive. The flow of the video playback is switched between these two states alternatively. In the normal state, clients retrieve the video from the broadcast channel for a period of time following an exponential distribution with rate of δ_p. In the interactive state, the user issues one of the interactive functions: PAU, JF, JB, FF and FR with the probability of p_1, p_2, p_3, p_4 and p_5 respectively such that $\sum_{i=1}^{5} p_i = 1$. The duration of the PAU/FF/FR and the jump distance of JF/JB are exponential distributed with rate of $\delta_{pau}/\delta_{ff}/\delta_{fr}$ and δ_{jf}/δ_{jb} respectively. It is further assumed that the playback rate of FF and FR (R) is twice of the normal playback rate. Table 1 summarizes the parameters of the system.

TABLE 1
Parameters of the simulation model

Parameter	Normal value	Range of values
L	2 hours	-
N	20	-
λ	0.02/s, 0.1/s	0.01/s-0.1/s
$1/\mu$	500s, 5400s	300s-5400s
R	2 times of normal playback rate	-
P_1, p_2, p_3, p_4 and p_5	0.2	-
$1/\delta_p$	30 minutes	-
$1/\delta_{pau}$	5 minutes	-
$1/\delta_{ff/fr}$	5 minutes	-
$1/\delta_{jf/jb}$	5 minutes	-
B_{up}	0.25/0.5/1.0 times of normal playback rate	-
Percentage of free-rider	0%	10% - 90%

We first look at the bandwidth requirement of the conventional approaches, i.e. ABM and GCM, for the central server as shown in Fig. 4. In the figure, each curve is represented by "*Model(x)*" where "*Model*=ABM or GCM" and "*x*=the departure time of client". It can be seen that the bandwidth requirement is increasing with the increase of arrival rate. For the same arrival rate, it can be found from the results that the system with free-rider requires less server resources than the system without free-rider. It is found that GCM can reduce the bandwidth requirement up to 12% compared to ABM.

We now focus on the performance of the proposed framework. In the following, each curve is represented by "*P2P(x, y, z)*" where "*x*=arrival rate", "*y*=peer's bandwidth" and "*z*=percentage of free-rider". The performance is measured in term of normalized bandwidth requirement $B_{normalized}$, where $B_{normalized}$ is equal to bandwidth required for the server in proposed approach over that in GCM. Fig. 5 first shows the performance of the system in various population of free-rider. From the result, it can be found that the bandwidth requirement is increased when the number of peers which intend not to share their resources to the system for VCR operations is increasing. In various scenarios, there is more than 95% of VCR operations can be completed in P2P manner when the population of free-rider is less than 50%. In addition, the system can perform better if the arrival rate is increased. It is because there are more peers provided to support VCR operations. On the other hand, if each peer can contribute more outbound bandwidth to the system, better performance can be resulted. It can be seen from the result that our proposed approach can reduce the server resources significantly compared to GCM.

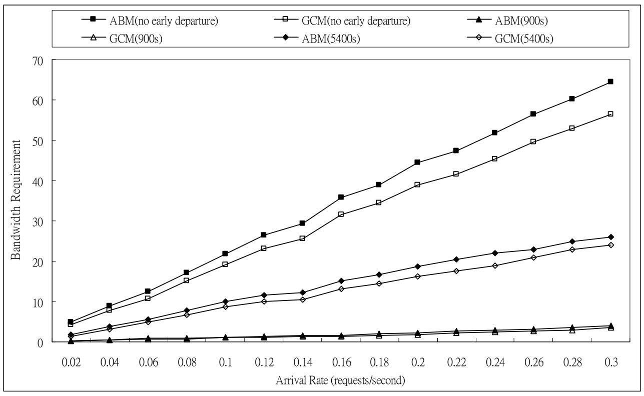


Fig.4 Bandwidth requirement against arrival rate

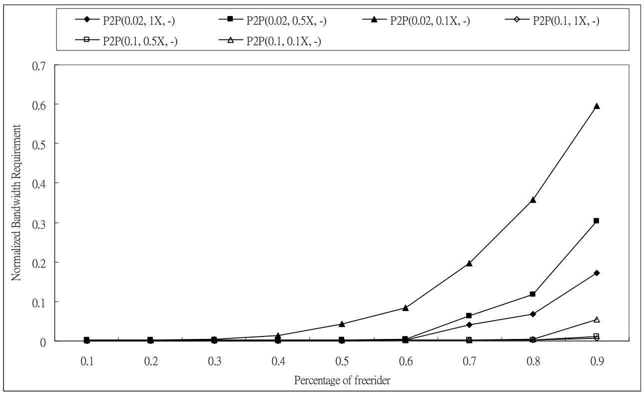


Fig.5 Bandwidth requirement against percentage of free-rider

Then, we look at how the departure rate affects the system performance as shown in Fig.6. From the result, it can be seen that the system performance can be improved if each peer can contribute more resources (i.e. outbound bandwidth) to the system in various departure rates. It can be found that the bandwidth requirement of the system with high arrival rate is reducing whereas that with low arrival rate is first increasing and then decreasing when mean time to departure (*MTTD*) is increased (i.e. the departure rate is decreased).

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

When both of the arrival rate and departure rate is high, the size of the P2P network for each segment is small that cannot fully satisfy the need of contingency channel to complete VCR operation and thus the server is required to release more resources. When the *MTTD* is further increasing, the size of the P2P network is enlarged and most of VCR operation can thus be done in P2P manner. In the case of low arrival rate, the need of contingency channel is low and thus the system requires less server resources. When the *MTTD* is increasing, there are more clients staying in the system and thus there are more VCR operations issued. However, the size of P2P network is still small which is not enough to serve the increased VCR operations and thus increases the need of server resources. As the *MTTD* is further increasing, the size of P2P network is large enough to serve the VCR operations.

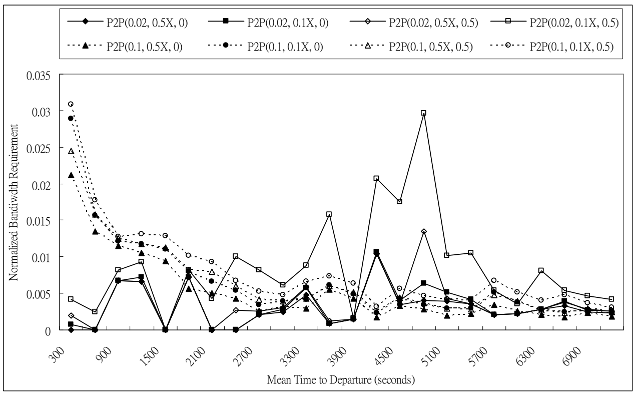
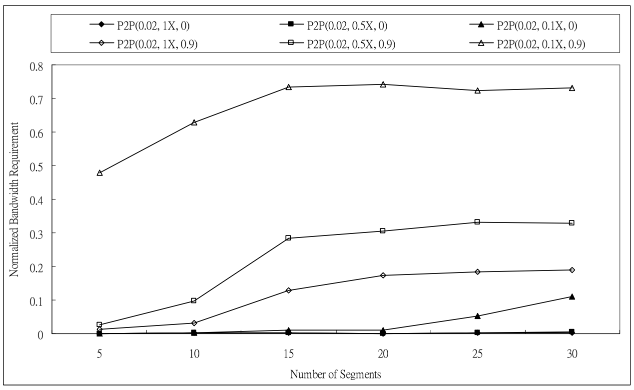
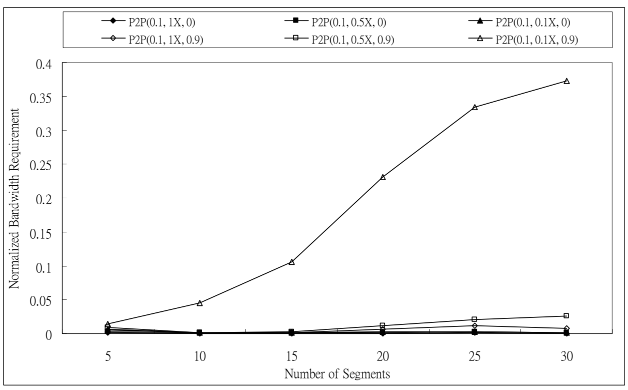


Fig.6. Bandwidth requirement against departure rate of peer



(a) arrival rate = 0.02



(b) arrival rate = 0.1

Fig.7. Bandwidth requirement against departure rate of peer

Finally, we examine the system performance in various batching time. Since the batching time can be computed according to the segment size, we can simply consider the number of segment are partitioned in the video. Fig.7 depicts the system performance against the number of segment in various arrival rates. The figure shows that the bandwidth requirement is increasing when the number of segment is increased. When the number of segment is increasing (or the batching time is decreasing), the size of the P2P network for each segment is decreased and thus fewer peers can be provided to serve each segment. On the other hand, the performance of the system with high arrival rate is better than that with low arrival rate. It is because the size of the P2P network for each segment is increased when the arrival rate is increasing. Although large segment size can obtain better performance, this increases the size of the client buffer and the waiting time as well. The system designer should consider the trade-off between the bandwidth requirement of the system and the client's buffer size.

V. Conclusion

In this paper, we investigate a possible solution for supporting VCR functionality in broadcast environment with peer-to-peer paradigm. Based on the GCM approach, we consider the use of the client's resources to assist the VCR operations. From the simulation results, this approach can significantly improve the system performance.

References

[1] K.A. Hua, M.A. Tantaoui and W. Tavanapong, "Video Delivery Technologies for Large-Scale Deployment of Multimedia Applications," *In Proc. of The IEEE*, vol. 92, no. 9, pp. 1439-1451, Sept. 2004.

[2] A. Ganjam and H. Zhang, "Internet multicast video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 159–170, 2005.

[3] K.M. Ho, W.F. Poon and K.T. Lo, "Video-on-Demand Systems with Cooperative Clients in Multicast Environment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 3, pp.361-373, Mar 2009.

[4] K.C. Almeroth and M.H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," IEEE Journal Selected Areas in Communications, vol. 14, no. 6, pp. 1111-1122, Aug. 1996.

[5] Z.M. Fei, M.H. Ammar, I. Kamel and S. Mukherjee, "Providing interactive functions for staggered multicast near video-on-demand," in Proc. of the IEEE Internation Conference on Multimedia Computing and Systems'99, pp. 949-953, 1999.

[6] W.F. Poon, K.T. Lo and J. Feng, "Performance study on implementation of VCR functionality in staggered broadcast video-on-demand systems," Journal of Systems and Software, vol. 75, no. 1-2, pp.95-107, Feb. 2005.

[7] J.B. Kwon and H.Y. Yeom, "Providing VCR functionality in staggered video broadcasting," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 1, pp. 41-48, Feb. 2002.

[8] A. J. Ganesh, A.-M. Kermarrec, and L. Massouli, "Peer-to-peer membership management for gossip-based protocols," *IEEE Trans. Computer.*, vol. 52, no. 2, pp.139-149, 2003.

[9] Y. Liu, L. Xiao, X.Liu, L.M. Ni and Z. Zhang, "Location Awareness in Unstructured Peer-to-Peer Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 2, pp. 163-174, Feb. 2005.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)