Proceedings of 2009 APSIPA Annual Summit and Conference, Sapporo, Japan, October 4-7, 2009

Cooperative Video Broadcasting Scheme with Scalable Video Coding

K.M. Ho and K.T. Lo
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
enktlo@polyu.edu.hk

Abstract—**In this paper, we propose a peer-based cooperative streaming framework with the use of broadcast capability of the network. The basic idea is to ask the concurrent high-end machines denoted peer server to cooperate to broadcast a video encoded by multiple description coding (MDC) to clients. This transmission strategy aims to take the benefit of peer-to-peer (P2P) and video broadcasting for video delivery in the heterogeneous network environment. Due to the dynamic nature of the P2P environment, the system performance of the proposed framework depends on the several factors such as the central server resource supplied, the number of peer server recruited and the minimum quality levels acquired. Hence, the main focus of this paper is to explore the relationship among these factors in the proposed architecture by an analytical model.**

I. INTRODUCTION

With recent advances in the technologies of high-performance networks and digital video compression, it is now possible to support Video-on-Demand (VoD) services for end-users. To provide a cost effective and scalable solution for a large-scale VoD system, a number of system topologies and transmission protocols have been proposed in the literatures, such as proxy caching, content delivery network (CDN), peer-to-peer (P2P), periodic broadcast, in the past decade. Proxy caching allows clients to fetch the cached video from the proxy directly so that the workload of the central server can be alleviated [1]. CDN is an extension of the proxy caching that video data is replicated to multiple CDN servers [2]. In P2P architecture [3], each end-point called peer which retrieves what it requests from the system and forwards/relays it to the system so that the bottleneck of the system is no longer at the server side. As the successful deployment of IP broadcast delivery [4], people have also exploited broadcast capability of the network to support large-scale video streaming services [6]. Nevertheless, these approaches still have their own problems for video delivery. Proxy caching and CDN are expensive to deploy and maintain. In P2P, the bandwidth requirement inside the network can be rapidly increased when more clients join the same video session. Broadcasting protocols such as Harmonic [7] are impractical to support insensitive start-up delay since the server needs to manage a large number of concurrent channels for a single video.

However, it can be noticed that P2P and broadcast approaches are the two most simplest and economical way to stream the video data over the network. It introduces us an interesting question if we can have a mechanism which can take either advantage to compensate either disadvantage. For these reasons, a number of pioneer protocols [8-11] are thus proposed to take either advantage (to compensate either disadvantage) by exploring the feasibility of using the broadcast capability of the network coupled with P2P transmission strategy for video transmission. Nevertheless, these studies are only focused on homogeneous network environment. In this paper, we propose a peer-based cooperative streaming framework with the use of broadcast capability of the network in the heterogeneous network environment. The basic idea is to ask the concurrent peer to cooperate to broadcast different quality levels to clients with heterogeneous bandwidths. In the proposed scheme, a number of high-end machines denoted peer server being willing to contribute their resources are recruited. Given that the video source is multiple description coding (MDC) bitstreams, each PS is responsible to broadcast one or more descriptions over the network. Note that each PS is also allowed to enter and leave the system at any time in our framework. On the other hand, it is assumed that clients can tolerate the degradation of the video quality provided that the system can guarantee to serve them with the minimum video quality. Thus, more than one PS or a little server resources will be injected into each description to increase the degree of reliability. The main focus of this paper is to show the relationship among the central server resources supplied, the number of peer server recruited and the minimum quality levels acquired in the proposed architecture by an analytical model.

In the following, the paper is organized as follows. We first describe the system architecture in Section II. The analytical model of the proposed architecture is developed in Section III. In Section IV, the results will be presented to demonstrate the efficacy of the proposed system. Finally, some concluding remarks are given in Section V.

II. SYSTEM DESCRIPTION

Consider a video with L seconds long which is streamed across a number of clients with heterogeneous inbound bandwidths over a broadcast-enabled network. To cater for the heterogeneous requirement, the system will encode the video into M descriptions, each of which is assumed to have the same rate of R bps and is transmitted over a separate broadcast channel. Generally, the quality of the video the

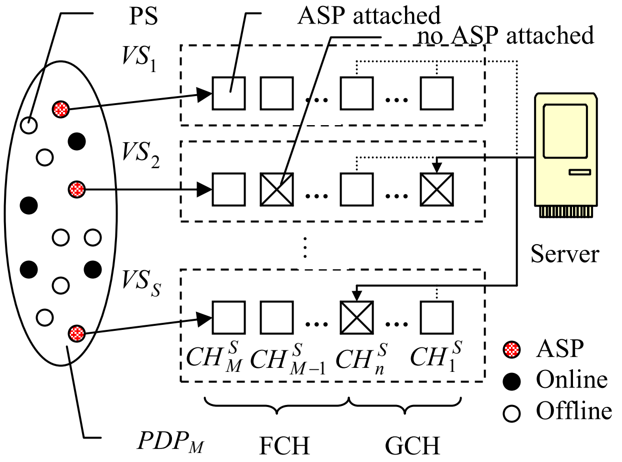

Fig. 1. The idea of the proposed framework

clients acquired depends on how many broadcast channels they join. That mean, the video quality is improving if more descriptions can be obtained. In the proposed framework, the system creates several *video sessions* (VS) for the video in staggered manner that each session is separated by W seconds. Thus, each VS comprises M broadcast channels to deliver M descriptions periodically. Denote VS_i and CH_j^i be the i^{th} video session of the video and broadcast channel for the j^{th} description of the video in VS_i respectively, i.e. $CH_j^i \in VS_i = \{CH_1^i, ..., CH_M^i\}$. As a result, the total bandwidth requirement of the system is $B = LMR/W$ and the number of video session in the system is $S = L/W$.

In client-server architecture, a single central server is required to allocate bandwidth of B for a single video. Obviously, in such architecture, the bandwidth requirement is linearly increased with the number of videos provided. In order to disperse the workload of the system so that the bottleneck of the system is no longer on the server side, a distributed scheme based on P2P paradigm is proposed. In the proposed scheme, a number of high-end machine denoted peer server (PS) being willing to contribute their resources such as bandwidth and storage are recruited. Each PS is responsible to handle a part of the duty of video broadcasting to deliver one or more descriptions. However, similar to other P2P applications, each PS in the system is allowed to leave and enter the system at arbitrary time. The service is then disrupted discontinuously between two states. It is obvious that this scene is unacceptable particularly in broadcast environment where thousands of clients are annoyed. Therefore, in order to increase the degree of reliability, more than one PS may be deployed for each description. PSs handling the identical video description form a *peer description pool* (PDP). Let PDP_j be the PDP handling the j^{th} description. VS_i randomly selects one of the online peers (what we calls it an *active serving peer* (ASP)) from the PDP_j to serve its CH_j^i. When the current serving PS (i.e ASP) goes offline, VS_i will pick another online PS from the PDP_j to carry on the service. Therefore, the corresponding video description can be kept on broadcasting. Occasionally, there is not enough online PS to be an ASP in PDP_j and thus the service of the corresponding $CH_j^i, \forall i$ is broken down. Thus, a central server is still deployed in the proposed framework to take over the service when any $CH_j^i, \forall i$ cannot own its ASP. But, it is also the case that clients can tolerate the degradation of the video quality provided that the system can guarantee to serve them with the minimum video quality. For example, the system can pledge to fulfill the promise it can support two quality levels of the video uninterruptedly. As a result, only G descriptions among M descriptions will be guarded by the central server. The broadcast channel (or description) supported by the central server is denoted *guarded channel* (GCH). The channel which does not require the involvement of the server resources is called *freedom channel* (FCH). The idea of the proposed framework is graphically illustrated in Fig 1. In the figure, VS_i is composed of M CHs (i.e. $CH_j^i, j = (1, M)$), each of which picks up an ASP from its corresponding PDP (i.e. $PDP_j, j = (1, M)$) to broadcast the dedicated description over the network. It can be indicated that the broadcast channel in the set $\{CH_i^j \mid i \in (1, n), j \in (1, S)\}$ is the GCH of the system and the one in the set $\{CH_i^j \mid i \in (n+1, M), j \in (1, S)\}$ is the FCH. In this scenario, we can see that CH_{M-1}^2, CH_1^2 and CH_n^S are the channels which cannot own its ASP. Because the first n channels are guarded by the central server, the duty of the CH_1^2, CH_n^S can be taken over by the server. However, CH_{M-1}^S is an FCH and thus it is ignored. Therefore, clients can only receive *M-1* descriptions from VS_2 whereas clients in VS_s can obtain all descriptions at this moment in time.

III. SYSTEM MODEL

In the proposed architecture, each PS is allowed to enter and leave the system at arbitrary time. It is first assumed that the mean up time and mean down time of each PS are independent and identically distributed with exponential function with the rate γ_{ON} and γ_{OFF} respectively. Then, the availability of the PS can be defined as $A = \frac{1/\gamma_{ON}}{1/\gamma_{ON} + 1/\gamma_{OFF}}$ [8]. In the proposed framework, several CHs for the same video description from different VSs share a group of PSs from their PDP. When they cannot own theirs ASP, they should wait until it can own so. Meanwhile, they have to request the assistance of the central server to carry on the service if they belong to GCH. In order word, it can be considered that there are K PSs in the PDP and n of them are online at time t. All CHs from this PDP can have theirs ASP if $n >= S$. Otherwise, $S - n$ CHs should wait on the queue. The CH will be queued (or served by another ASP) immediately when its current ASP goes offline. Therefore, the number of

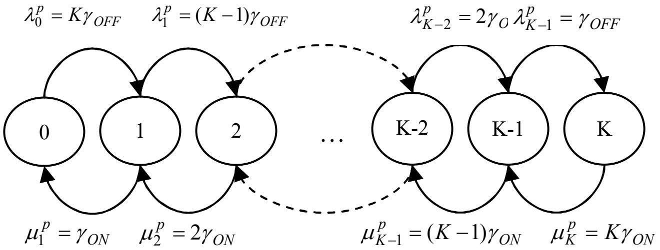

Fig. 2. The CTMC of the PDP

online PSs in the PDP determines the waiting time of the CH as well as the requirement of the central server resources. We use Continue-Time Markov Chain (*CTMC*) [7] to model the behavior of the PDP whose state-space diagram is shown in Fig 2. Denote λ_k^p and μ_k^p be the transition rate from state *k* to state *k+1* and from state *k* to *k-1* respectively. When the system is at state *0*, there is no PS to be an ASP and it is implied that all CHs should wait on the queue. State *K* here denotes that all PSs in PDP go online. Let p_n be the probability of *n* online PSs in the PDP. Then, the flow balance equation can be expressed as

$$p_{n+1} = \left(\frac{\lambda_n^p}{\mu_{n+1}^p} \right) p_n \tag{1}$$

Using eqn(1) yields

$$p_n = \binom{K}{n} r^n p_0 \tag{2}$$

, where $n > 0$ and $r = \gamma_{OFF} / \gamma_{ON}$. The algebraic form of the $\{p_n\}$ does not allow an easy closed-form calculation of p_0 . Instead, we first compute each of the coefficients in front of p_0 . That is, let a_n be the coefficient of p_0 and so we have $p_n = a_n p_0$. Since $\sum_{i=0}^{K} p_n = 1$, we have

$$p_0 = \left(1 + \sum_{i=1}^{K} a_i \right)^{-1} \tag{3}$$

As we have known that max(*S*-*n*, 0) CHs should wait on the queue for an ASP when there are *n* online PSs in the PDP. Thus, the number of video session and the number of PSs in the PDP settle the average queue length. To find the average number of the CHs in the queue ($\overline{L}$), we use the definition of expected value and obtain

$$\overline{L}(x,y) = \begin{cases} \sum_{i=0}^{x} (y-i)p_i = p_0 \left(y + \sum_{i=1}^{y-1} (y-i)a_i \right), y \le x \\ \sum_{i=0}^{x} (y-i)p_i = p_0 \left(y + \sum_{i=1}^{x} (y-i)a_i \right), y > x \end{cases} \tag{4}$$

, where *x, y* represent the parameters of the number of PSs in the PDP and the number of video session respectively. Accordingly, for the average waiting time of the CH ($\overline{W}$), we know from Little's formula that

$$\overline{W}(x,y) = \frac{\overline{L}(x,y)}{\gamma_{ON}(x - \overline{L}(x,y))} \tag{5}$$

During waiting for the ASP, the duty of the CH (i.e. GCH) without ASP should be supported by the central server. On the other hand, the service time of the ASP for the CH with ASP lasts for $1/\gamma_{ON}$ on average. Thus, the average bandwidth required by the central server for the PDP_i (V_i) is given by

$$V_i\ (x,y) = \frac{R \cdot \overline{W}(x,y)}{1/\gamma_{ON} + \overline{W}(x,y)} \tag{6}$$

Denote D_i be the number of PSs in PDP_i and G be the number of GCH. Then, the overall bandwidth required for the central server is computed by

$$B_{server} = S \cdot \sum_{i=1}^{G} V_i\ (D_i\ , S) \tag{7}$$

As mentioned in Section II, some of the broadcast channels are FCH and thus clients need to tolerate the degradation of the video quality. Obviously, the average video quality provided depends on the availability of each FCH in the same video session. Let A_i^j be the availability of the CH_i^j (i.e. the probability of the CH_i^j that can own its ASP), which can be defined as

$$A_i^j(x,y) = \frac{1/\gamma_{ON}}{1/\gamma_{ON} + \overline{W}(x,y)} \tag{8}$$

, where *x, y* represent the parameters of the number of PSs in the PDP_i and the number of VS respectively. Hence, the expected number of descriptions received from FCH in the same VS (*J*) can be computed by

$$J = \sum_{z=1}^{M-G} z\mathrm{P}(\text{any z descriptions received from FCH})$$

(10)

Thus, the average video quality received ($\overline{Q}$) is equal to $\overline{Q} = G + J$.

IV. RESULTS

In this section, we evaluate the performance of the proposed framework. It is assumed that the video has the length of 7200 seconds long with the rate of *R*=1 and is encoded into 10 quality levels (i.e. *M*=10 and 10 PDPs). Each video session is separated by 600 seconds and thus we have *S*=12. We assume that each PS with $\gamma_{ON} = 36000A$ seconds and $\gamma_{OFF} = 36000(1-A)$ seconds contributes a bandwidth of *R* bps. Therefore, the online/ offline time of each PS is ranged between 1 to 10 hours governed by *A*. In addition, unless other specified, *G*=10. Finally, for simplicity, it is also assumed that all PDPs have the same parameters.

We first investigate how the number of PSs affects the system performance. Fig. 3 plots the total server bandwidth

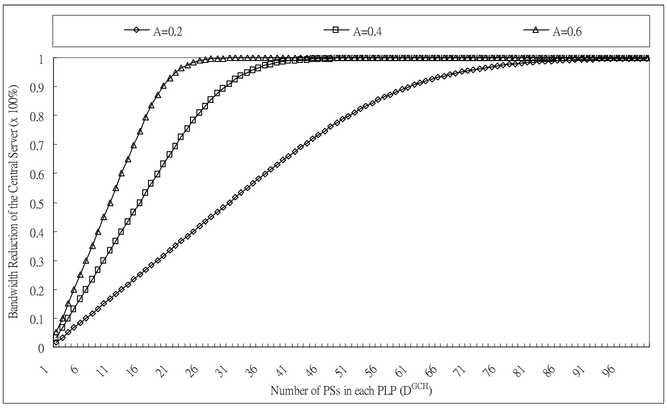

Fig. 3. Number of PSs against Server Bandwidth Reductions

reduction (i.e. $\frac{B - B_{server}}{B} \times 100\%$) versus the number of PSs per PDP in various availabilities of PS. It can be observed that the server bandwidth reduction is first increasing when the number of PSs is increased. Then, this reduction is not significant for further increasing the number of PSs to each PDP. It can also be found that the availability of the PS appear to be another important variables in the system. For the same level of reduction, higher availability requires fewer PSs in each PDP. For example, when the central server has a target of the bandwidth reduction of 10% for this video, the number of PSs required for the case A=0.2, A=0.4 and A=0.6 are 19, 30 and 61 respectively.

Then, we look at the expected number of quality levels (i.e. $\overline{Q}$) that the clients can enjoy when the number of GCH (i.e. G) and the number of PSs in each PDP are changed in various availabilities. It can be found from the Fig. 4 that $\overline{Q}$ is improving when the number of PSs and its availability are increased. It is because the probability of each video channel owned its ASP is increased and thereby increase the opportunity that clients can obtain more descriptions. For the same quality level, we can see that the number of PSs in each PDP is decreased when G is increased. For example, in order to acquire seven quality levels, each PDP needs to recruit 18 PSs if A=0.4 and G=1 but it only requires 10 if A=0.4 and G=5. It can be evaluated that this difference is come from the different contributions of the central server resource. Therefore, system designer should consider the tradeoff between the central server resources as well as the number of PS to the quality levels obtained by the clients.

V. CONCLUSION

In this paper, we develop a possible solution for building a VoD system using video broadcast coupled with P2P paradigm. The basic idea is to ask the concurrent peer server to cooperate to broadcast a video encoded by MDC to clients. An analytical model is developed to explore the several system parameters that can influence the system performance. It can be found from the results that the workload of the central server and the acquirement of the quality levels are improving when the number of PSs and its availability are increased. In addition, it can be seen that the systems requires more PSs and server resources when the minimum number of the quality level required is increased.

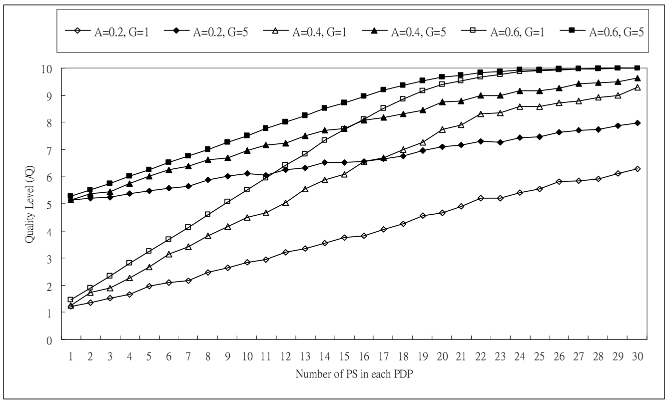

Fig. 4. Number of PSs against Quality Levels

REFERENCES

[1] J.C. Liu and J.L. Xu, "Proxy caching for media streaming over the Internet," IEEE Communications Magazine, vol. 42, issue 8, pp. 88-94, Aug. 2004.

[2] A. Biliris, C.D. Cranor, F. Douglis, M. Rabinovich, S. Sibal, O. Spatscheck, and W. Sturm. , "CDN brokering," J. Comput. Comm., vol. 25, no. 4, pp. 393-402, Mar. 2002.

[3] W.P. Yiu, X. Jin and S.H. Chan, "Challenges and Approaches in Large-Scale P2P Media Streaming," IEEE Communication Magazine, pp. 50-59, 2007.

[4] A. Ganjam and H. Zhang, "Internet multicast video delivery," Proceedings of the IEEE, 93(1):159–170, 2005.

[5] 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.

[6] L. Juhn and L. Tseng, "Harmonic broadcasting protocols for video-on-demand service," IEEE Trans. on Broadcasting, vol. 43, no. 3, pp. 268-271, 1997.

[7] A. Hoyland and M. Rausand, "System Reliability Theory: Models and Statistical Methods," John Wiley and Sons, vol. 518, 1994.

[8] K.-K. To, J. Lee and S.-H. Chan, "Decentralized periodic broadcasting for large-scale video streaming," in Proceedings of International Con-ference on Multimedia & Expo (ICME), pp. 285-288, Amsterdam, The Netherlands, 6-8 July, 2005.

[9] X. Yang, P. Hern´andez, F. Cores, A. Ripoll, R. Suppi, and E. Luque, "Distributed p2p merging policy to decentralize the multicasting delivery.," In 31st EuroMicro Conference on Software Engineering and Advanced Applications, pp. 322–329, Porto, Portugal, Aug.-Sept. 2005.

[10] K.M. Ho, W.F. Poon and K.T. Lo, "Peer-to-peer batching policy for video-on-demand systems," Proceedings IEEE First International Conference on Communications and Networking in China 2006 (Chi-naCom '06), pp. 1-6, Oct. 2006. Beijing, China.

[11] K.M. Ho, W.F. Poon and K.T. Lo, "Video-on-Demand Systems with Cooperative Clients in Multicast Environment," accepted for publication, IEEE Transactions on Circuits and Systems for Video Technology, 2009. (to be published)