Scheduling Strategy for Realistic Implementation of Video on Demand over IPTV System

LIN-HUANG CHANG[1], MING-YI LIAO[2], JIUN-JIAN LIAW[1]
[1]Grad. Inst. of Networking and Communication Eng., Chaoyang University of Technology
[2]Dept. of Computer Science and Information Eng., Chaoyang University of Technology
168, Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan

Abstract: The tradeoffs in employing multicasting technology for implementing large scale video on demand (VoD) over Internet Protocol Television (IPTV) system, in general, come to the scheduled multicasting streams and limited video or television program selections. The scheduling strategies presented in this paper will provide several advantages. First, we proposed a near Video on demand architecture combining multicasting and unicasting batching with no scheduled multicasting stream. Second, we dynamically adjust the admission threshold to alleviate the server loads and improve bandwidth utilization. The idea of virtual community with the introduction of the chatting robots in our system is constructed to fulfill the dynamic adjustment of the batching time.

Key-Words: multicast streaming, IPTV, VoD, virtual community

1 Introduction

The advances in computers and communication technologies such as digital video streaming and high speed networks during the last decade have made Internet Protocol Television (IPTV) service feasible [1-2]. IPTV describes a system where a digital television or video service is delivered to consumers using the Internet Protocol (IP). IPTV covers both live TV and stored video through multicasting or unicasting. The TV programs and videos are usually requested upon customers' demands. Therefore, through the cable TV or Internet, the geographically distributed customers are allowed to request and view high quality videos from the multimedia server complexes. It is needed to ensure that the customers have an overall video quality of experience at least as enjoyable for the video on demand over IPTV system [3]. With the use of MPEG standard, the VoD servers not only store and retrieve a great volume of compressed video data, but also meet the real time requirement of video data which is transmitted to several viewers over broadband networks.

The commercialized VoD systems, including Apple QuickTime 7, Real Network RealSystem, Cisco IP/TV, Microsoft Media Play, Streaming 21 Showcases IPTV, and so on, basically employ real-time transport protocol (RTP) and real time streaming protocol (RTSP) to deliver digital multimedia data from the streaming server. Digital video is displayed and then discarded once the viewers have watched it. The technologies which have been implemented usually handle all industry standard formats such as high quality MPEG-2 and MPEG-4 videos as well as MP3 audio. Typically, each VoD server is capable of supporting hundreds of stored digital videos and delivering more than hundreds of high quality streams simultaneously. Load balancing between multiple servers, network cards and processors is also provided for these VoD systems. The bandwidth support usually covers from 28.8 Kbps to 16 Mbps. The typical bandwidth requirements to view high quality MPEG-2 and MPEG-4 digital videos can be as high as 6 Mbps. The VoD systems have been used in the areas of distance learning, video rental service, videoconferencing, and home shopping, etc.

However, there is always a tradeoff between the number of video selections and available resources, such as channel utilization and server loads [4]. In this research, we designed a scheduling strategy to realize a large scale VoD system by combining multicast with unicast streams. The ideas of dynamic batching time and virtual community using chatting robots are also proposed in this paper to solve the resource under dense usage.

The rest of this paper is organized as follows. Section 2 gives a brief review of the related works in implementing VoD system. Section 3 presents the proposed VoD architecture combining multicasting and unicasting batching. The scheduled strategies using dynamic adjustment of batching time are reported is Section 4. The idea of virtual community with chatting robots is proposed in Section 5 to complete the VoD system. This paper concludes with Section 6.

2 Related works in implementing VoD system

The implementations of the VoD services over high speed networks or cable TV networks have been discussed previously [5-9]. In [6], the authors reported their progress in supporting delivery of MPEG-2 audio/video streams over various types of high-speed networks, e.g. ATM, Ethernet and wireless. The research in this testbed included video transmission with heterogeneous Quality of Service (QoS) provision, variable bit rate (VBR) server scheduling and traffic modeling, as well as video transmission over the Internet and IP-ATM hybrid networks. In [7], the authors implemented the VoD system at National Tsing Hua University in Taiwan by buffering the incoming streams at the intermediate routing nodes. The VoD tried system over a cable TV infrastructure, was also implemented in the Science-Based Industrial Park, Hsinchu, Taiwan. More than two hundred household TVs connected to the network through the set-top-boxes (STB) were served over that system.

In addition to the commercialization and implementation of VoD systems discussed above, many researches continuously work on the design of high performance VoD systems capable of handling large numbers of services simultaneously over geographically large scaled internet. While the true VoD system provides a very short latency it might be inappropriate for implementing large scaled VoD services. The major problems of the true VoD system are the dedicated transmission channel for every single customer and the large access loading to the VoD servers. This results in high cost to customer and a low service rate or a long waiting queue on the unicast channels for the true VoD systems.

On the other hand, near VoD systems make the large scaled VoD implementation realistic by employing multicast streaming technology. Several researches [10-16] have worked on the various batching policy, channel allocation and admission control for VoD using multicasting and unicasting technologies. Improved performance in service rate and short latency were achieved by researches [14-16]. The double-rate batching policy [14] was proposed to reduce the start-up delay of VoD service. The tradeoffs of these VoD systems were scheduled multicasting for one single video all the time and consequently result in the limited video selections. This resulted in inefficient usage of the transmission bandwidth no matter how frequent the customers request for the same video.

The key issue in applying large scaled VoD systems, however, is cost rather than technology. In VoD system, it is expected for the server to get as many isochronous requests as possible for the same streaming video. It is common for the popular videos, especially. The popular videos are defined to be the videos being requested simultaneously or at a short interval by various customers. Usually, the majority of people requests for a video at the same period such as evening time after work or weekends. The popular videos are not only because of their popularity but also because of the similar leisured schedule of human being. Therefore, it is important to apply a dynamic batching scheme when we implement the VoD system for a large scale.

In this study, we propose a VoD system combining multicasting and unicasting batching to reduce the latency. Moreover, with scheduled multicasting, the proposed architecture provides better efficiency in the transmission bandwidth usage and more selection in the number of the videos. The batching time of the proposed VoD architecture is dynamically adjusted. The adjustment depends on the time zone when the customers request for videos. Five different time zones with dynamic adjustment of batching time will be discussed to reduce the server loads and unnecessary waiting latency as well as to increase the efficient usage of the transmission bandwidth. The virtual community (VC) with chatting robots is created to keep customers from impatient logout during the waiting time.

3 Architecture of VoD system combining multicast and unicast batching

Figure 1 illustrates the architecture of the proposed VoD batching system. There are totally N channels, of which N_M are multicast channels and N_U are unicast channels. We assume the network bandwidth at the client side can accommodate two video streaming simultaneously. Also, minimum local buffer size is required to run the proposed near VoD system.

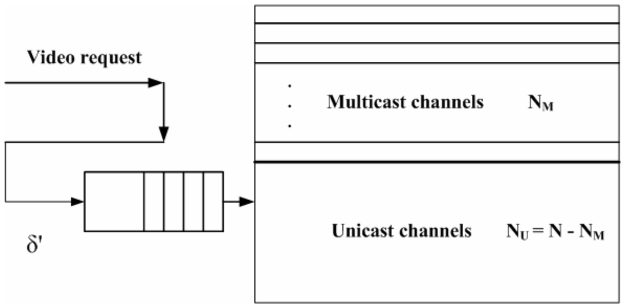

Fig.1 The architecture of VoD batching system

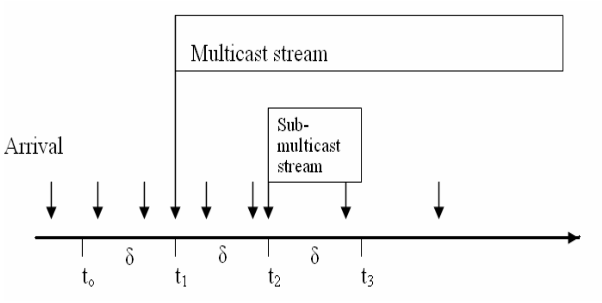

Fig.2 The proposed batching timing

The proposed batching timing is shown in Fig.2. When a request arrives at time t, the system checks the database in the server first to see if there is any request with the same video on queue of multicasting. If there is no same request before, the system initiates a multicast queue at time t=t_0, and starts the multicast stream after δ seconds, where δ is the admission threshold of waiting queue which is also a tolerable waiting time for customers being logout. If there is already a multicast queue being initiated and the arrival time t is in between t_0 and t_1, the request will be put into the multicast queue and starts the multicasting after (t_1-t) seconds. However, if there is a request arriving at time between t_1 and t_2, where t_2-t_1=δ, the customer will cache the playing multicast stream right away. Then, the VoD system will initiate another multicast stream called sub-multicast at time t_2 for those customers arrive between t_1 and t_2. In this case, the client side needs to cache the multicast video data with a time period equals to (t-t_1) seconds and experience the play latency with (t_2-t) seconds. The sub-multicast stream is released after (t-t_1) seconds and thereafter the client will continue video play back from the cached multicast stream being stored locally.

After time t_2, no more multicast stream will be initiated for the same batching. In stead, if the request arrives between t_2 and t_3, where t_3-t_2=δ, the system will put the request into the unicast queue and cache the multicast stream, starting at time t. The waiting queue δ' at the unicast channels (shown in Fig. 1), will be the latency in this request. In the worst cast, the video client needs a cache to store video data up to a time period of 2δ seconds. The waiting queue δ' at the unicast channels, will be the latency of this request. The unicast channel can be released after a time period of (t_3-t_1) seconds. If the request arrives even after time t_3, the system will initiate a new multicast queue which is different the previous streams.

From our batching system, the latency experienced by a customer depends on when the customer requests for a video, i.e. the arriving time. For the client requests the video at time between t_0 and t_2, the average latency of this VoD system, in principle, is δ/2, assuming the multicasting channels are available all the time for the batching system. However, the average latency of our architecture will be δ'/2 for the arriving time between t_2 and t_3. This latency is simply the time cost of the unicast queue. The algorithm of the batching scheme is listed in Table 1.

Table 1. The batching scheme

```
If (queue requests are in the multicast queue within 2δ seconds) then

  If (arriving time $t \le t_1$) then
    Wait in the multicast queue ;
  Else if (arriving time $t \le t_2$) then
    Cache the data of the multicast stream ;
    Wait in the sub-multicast queue ;
  Else if (arriving time $t \le t_3$) then
    Cache the data of the multicast stream ;
    Wait in the unicast queue ;
  Else
    Initiate a new multicast queue at arriving time t ;
    Set $t=t_0$ ;
  Endif

Endif
```

Intuitively, we can vary the number of multicast channels N_M (unicast channels N_U) from zero (N) to N (zero). With N_M=zero, the batching system will assign every request a unicast channel. It becomes a true VoD system. However, the blocking probability for the request will be significant high fo a large scaled VoD system. With N_M=N, the waiting queue and the threshold δ of the multicast channels will be minimized. However, it becomes a near VoD system even when the requested frequency of the video is low. In that case, the average waiting time for the play back after request will increase.

Taking the approximation as discussed in reference [20] with M/G/k/k queueing system, it is possible and fair to obtain the average latency of our VoD system as δ/2, i.e. δ=δ'. Comparing to other near VoD and true VoD systems [12-16], our proposed system can provide a better serving rate with the same average latency for customers. The flexibility of selecting video is another contribution from our near VoD system due to the free timetable of the multicast streams. The local video client, however, is required to store up to 2δ seconds of video data. If the video is encoded as MPEG-1 with average rate of 1.5 Mbps, a local client with 56.25 MB buffer is needed to batch video up to 300 seconds. From our near VoD system and other researches [12-16], the average tolerable latency is on the order of minutes (e.g. 5 min.).

Unfortunately, due to the similarity of the spare time of human being and the difference in the popularity of the requested video, it is not practical to model the VoD system simply using single approximation. Therefore, it is important to apply a dynamic batching scheme when we implement the VoD system realistically. The following section will give detailed discussion of the proposed strategies.

4 Scheduling strategy using dynamic batching time

In addition to the issues of waiting latency as well as bandwidth usage and local buffer requirement, the customer blocking is another important factor for the design of VoD system. The loading of the server and the usage of the transmission bandwidth are two basic issues of the blocking probability. They are mostly contributed by the request for popular videos at hot time zone. The requests for popular videos result in high demands of server loading. The hot time zone is defined as a period with dense customers which cause bandwidth request to increase. The cold time zone, on the other hand, is defined as low demands of server loading and less usage of the transmission bandwidth.

Therefore, it is not necessary to put the requests in the normal waiting queue, as discussed in the previous section, during the cold time zone. From this point of view, our VoD system will dynamically adjust the admission threshold from the minimum threshold, δ_{min}, to the maximum threshold, δ_{max}, depending on the arriving time zones. The idea of minimum threshold has been pointed out from the previous research, proposed by Almeroth [17] and impatient customer statistic results [19]. However, the complete implementation algorithm has not been reported yet. The minimum threshold in our system is expected to fulfill the true video-on-demand nature. It is true especially for the less popular video at cold time zone. Under such circumstance, the chance to gather two or more requests for the same video at the same batching period is low. Therefore, it should be a smart strategy to minimize the latency for the customers by providing the unicast service.

On the other hand, the maximum threshold will reduce the blocking probability of requested customers dramatically. This is especially possible for the popular videos being requested during the hot time zone. Under this situation, it is expected for the server to collect as many requests as possible within one batching time. The longer the admission threshold is, the more requests the same batching collect. Maximizing the admission threshold will benefit the system utilization a lot in this case. However, it should not detract from the on-demand nature.

According to the leisured schedule of human being and the statistic results from requests, we dynamically adjust our batching system to five different time zones, which are sets of time intervals. The first one is hot time zone (HT), e.g. sever to ten o'clock in the evening daily or ten to twelve o'clock in the morning as well as two to five o'clock afternoon during the weekend. The second zone, called sub-hot time zone (SHT), could be distributed around one to two hours before or after the HT. Normal time zone (NT) usually is the time period of one hour before or after people's work hours. There will be no adjustment of the original admission threshold and batching scheme at NT. Sub-cold time zone (SCT) is defined as the forth time zone which is roughly around one to two hours before sleep or after wake up. The last one is the cold time zone (CT) which covers most of the deep night period. The algorithm of our dynamic adjustment of the batching time is shown in Table 2. The arrival time is assumed to be t. The spirit of the dynamic adjustment is explained below.

As discussed early, it is not necessary to keep customers waiting for several minutes at cold time zone. Since the loading of server and the usage of transmission bandwidth is low in general for both popular and less popular videos. During sub cold time zone, we keep the requests for the popular videos with average queuing latency to increase the efficient usage of the transmission bandwidth. Otherwise, the admission threshold is adjusted to be δ_{min}. The admission threshold is back to the average queueing latency at the normal time zone for all provided videos. At sub hot time zone, the admission threshold is increased to δ_{max} to release the increasing requests for popular videos. Otherwise, the admission threshold keeps the same as original shceme. Finally, during hot time zone, we assign the admission threshold for the popular videos to be δ_{max} and will not provide the less popular video during this period. It is understood that the popular videos usually group much more requests at one batching than the less frequent ones. It is also suggested from Zipf's research [18] and movie rental statistics [19] that a small amount of the movie offerings at one time, especially the popular ones, will experience the largest requested volume. Obviously, in this case, every stream for the popular videos will gather much more requests due to the maximized admission threshold and increased chances for requesting popular videos. Therefore, compared to the action taken during SHT, the loading of server is expected to be further reduced. It should be noted that at HT the service rate is increased dramatically by removing the service of less frequent videos and maximizing the admission threshold.

Table 2. Dynamic adjustment of the batching time

```
If ($t \in HT$) then
    Adjust the admission threshold to $\delta_{max}$ ;
    Redirect the new arrival to the VC server ;

    If (the requested VC chatting room of the same video does not exist) then
        Create a new VC chatting room ;
        Generate a random number RN ;
        Dispatch RN VC robots to the new VC chatting room ;
    Else
        The new client joints the existing VC chatting room ;
    Endif

Else if ($t \in SHT$) then

    If (the requested video belongs to the popular video) then
        Adjust the admission threshold to $\delta_{max}$ ;
    Else
        Keep the original admission threshold $\delta$ ;
    Endif

Else if ($t \in NT$) then
    Keep the original admission threshold $\delta$ ;

Else if ($t \in SCT$) then

    If (the requested video belongs to the popular video) then
        Keep the original admission threshold $\delta$ ;
    Else
        Adjust the admission threshold to $\delta_{min}$ ;
    Endif

Else
    Adjust the admission threshold to $\delta_{min}$ ;
Endif
```

5 Architecture of virtual community

In order to prevent the logout of the impatient customers for the cases with maximum threshold, the virtual community (VC) is further proposed to extend the tolerable latency of the customers. The system manger will redirect the customers to the VC server while putting the request to the streaming server. The architecture of the VoD system with VC is shown in Fig. 3.

The VC server will assign an existed VC chatting room or create a new one for the customer depending on the existence of the requested video. This proposed strategy is also shown in the algorithm of Table 2. The VC chatting room will provide a brief introduction and summary of the video. The open discussing of the video can be proposed by chatting robots or customers as well.

The VC chatting robots are designed in the VC chatting room to energize the chatting activity and to propose interesting topics of the video to customers. A random number RN of VC chatting robots is assigned by the VC server automatically to prevent customers from discovery. The design of the VC chatting robot in the VC chatting room is also shown in Fig. 3.

To provide a better role play for the whole proposed system, the VC chatting robot with artificial intelligence (AI) is under developed. The full implementation of VC chatting rooms with chatting robots for all time zones is also constructed.

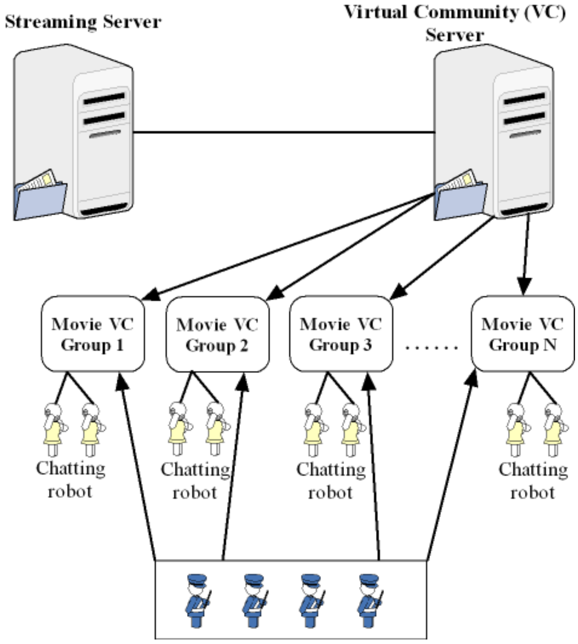

Fig.3 The architecture of VoD system with VC

6 Conclusions

The scheduling strategy proposed in this paper is believed to provide a very good batching scheme for the realistic implementation of the VoD system. We summarize our contribution as follows.

First of all, with no scheduled multicasting streaming, we have proposed a near VoD architecture combining multicasting and unicasting batching. More flexibility in video selections with short latency is provided in our VoD batching system.

Secondly, to solve the issues of server loading and bandwidth usage in VoD realistic implementation, we proposed the dynamic adjustment of the admission threshold in our batching system. Depending on the time zone of the arrival and the popularity of the requested videos, the batching scheme with five different time zones was designed to increase the efficient usage of the transmission bandwidth, especially for the request for popular videos, and to reduce the unnecessary waiting latency, especially at cold time zone.

Thirdly, the virtual community (VC) controlled by the VC server was constructed to fulfill the dynamic adjustment of the batching time. The appearance of VC will reduce the chance of the impatient logout with maximized admission threshold.

Finally, the scheduled strategy is completed by the introduction of the chatting robot in the virtual community. The VC chatting robot is designed to strengthen the functionality of the VC and to energize the VC chatting room.

Acknowledgement

The authors would like to thank the discussion and suggestion from Dr. K.C. Lai. This study is supported partially by the National Science Council of Republic of China, Taiwan under contracts, NSC 95-2221-E-324-020 and NSC 95-2622-E-324-013-CC3.

References:

[1] K. Kerpez, D. Waring, G. Lapiotis, J.B. Lyles, and R. Vaidynathan, "IPTV service assurance ," IEEE Journal in Communications Magazine, Vol. 44, No. 9, pp. 166-172, 2006.

[2] W. Park, C. Choi, Y.K. Jeong, "An Implementation of the Broadband Home Gateway supporting Multi-Channel IPTV, " Proc. of IEEE International Symposium on Consumer Electronics, pp. 1-5, 2006.

[3] V. Tokekar, A.K. Ramani and S. Tokekar, "Analysis of Batcing Policy in View of User Reneging in VOD System ," Proc. of IEEE INDICON, pp. 399-403, 2005.

[4] L. Souza, A. Ripoll, X.Y. Yang and P. Hernadez, "Designing a Video-on-Demand System for a Brazilian High Speed Network ," in 26th International Conference on Distributed Computing Systems, pp. 43-50, 2006.

[5] P. Christian, "IPTV gets personal", IEEE Journal in Communications Engineer, vol.4, no.4, pp.26-27, 2006.

[6] Y.H. Chu, S. Rao S. Seshan and H. Zhang, "A case for end system multicast", IEEE Journal in Communications, vol. 20, no. 8, pp. 1456-1471, 2002.

[7] S. F. Chang, A. Eleftheriadis, D. Anastassiou, S. Jacobs, H. Kalva, and J. Zamora, "Coolumbia's VoD and multimedia research testbed with heterogeneous network support," Journal on Multimedia Tools and Applications, Kluwer Academic Publishers, 1997.

[8] T. Su and J. Wang, "Buffered multicast routing for video-on-demand systems," 1999 IEEE International Conference on Communications, pp.1000-1004 1999.

[9] Dynamic service aggregation for interactive information delivery over networks. http://hulk.bu.edu/projects/summary.html

[10] C. Bouras, V. Kapoulas, A. Konidaris and A. Sevasti, "A dynamic distributed video on demand service," in 20th International Conference on Distributed Computing Systems, pp. 496-503, 2000.

[11] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," Multimedia System, pp.112-121, 1996.

[12] K. Han, J. H. Kim, Y. H. Won, "SRS: a viewer scheduling strategy using client's buffer in video-on-demand systems," IEEE TENCON, pp. 317-320, 1999.

[13] T. C. Su and J.S. Wang, "Buffered multicast routing for video-on-demand systems," IEEE ICC'99, pp.1000-1004, 1999.

[14] W.F. Poon, K. T. Lo and J. Feng, "Batching policy for video-on-demand in multicast environment," Electronics Letters, pp.1329-1330, July 2000.

[15] W. F. Poon, K. T. Lo and J. Feng, "Design and analysis of multicast delivery to provide VCR functionality in video-on-demand systems," 1999 2nd International Conference on ICATM, pp. 132-139, 1999.

[16] J. Y. B. Lee, "UVoD: an unified architecture for video-on-demand services," IEEE Communication Letters, Vol. 3, No.9, pp. 227-279, September 1999.

[17] K. C. Almeroth and M. H. Ammar "The use of multicast delivery to provide a scalable and interactive video-on-demand service," IEEE Journal on Selected Areas in Communications, Vol. 14, pp. 1110 -1122, 1996.

[18] G. Zipf, Human behavior and the principle of least effort. Reading, MA: Addison-Wesley, 1994.

[19] J. Yoshida, "The video-on-demand demand: opportunities abound, as digital video becomes a reality", Electronics Eng. Times, Mar. 15, 1993.

[20] A. O. Allen, Probability, Statistics, and Queueing Theory with Computer Science Applications 2nd ed. New York: Academic, 1990.