

Short-Delay Video Streaming with Restricted Supplying Peer Bandwidth

Hung-Chang Yang¹, Hsiang-Fu Yu¹, Li-Ming Tseng¹, Yi-Ming Chen²

¹Dep. of Computer Science & Information Engineering, National Central University,
Jung-Li, Taiwan

{cyht,yu}@dslab.csie.ncu.edu.tw, tsenglm@csie.ncu.edu.tw

²Dep. of Information Management, National Central University, Jung-Li, Taiwan
cym@im.mgt.ncu.edu.tw

Abstract. With the growth of bandwidth, real-time video streaming service becomes popular. Such application is also considered a future killer application on Internet. Recent research efforts have demonstrated the promising potential of building cost-effective video streaming systems on top of peer-to-peer (P2P) networks. Since the peers have limited capacity, such as upstream bandwidth, each streaming session may involve multiple supplying peers. In this paper, we propose a novel strategy to retrieve a long-duration video from multiple peers which have arbitrary and restricted upstream bandwidth, such that the waiting time is minimized. In comparison with the previous work [13], our strategy can greatly improve the waiting time. In the arbitrary given examples, our strategy can improve the waiting time by 67%. To take into account the popular compressed video with variable bit rate, we also show how to apply our strategy readily to the VBR videos

1 Introduction

With the advancement of broadband networking technology, and the growth of processor speed and disk capacity, real-time video streaming service is getting increasingly popular among users and contributes a significant amount of today's Internet traffic. However, there are still many challenges towards building cost-effective, robust and scalable video streaming systems due to the huge size, high bandwidth and delay requirements for video streaming.

A majority of video streaming architectures follows a client-server design. The server may have as many streams of each program as the current number of potential viewers (i.e. users currently connected to the service). Since the growth in bandwidth can never keep up with the growth in the number of viewers. It may easily run out of bandwidth and result in tremendous demand for communication bandwidth on the system and underlying network. To alleviate the stress on the bandwidth, there are many researches have been stimulated in the recent years. One way is to broadcast popular videos [8-11, 14, 15]. According to [6], 80% of demands are on a few (10 or 20) very popular videos. Because the server's broadcasting activity is independent of the arrivals of requests, the approach is appropriate to popular or hot videos that may

interest many viewers at a certain period of time. These approaches are using IP multicast technique to achieve serving multiple viewers using the same stream. However, the IP multicast has not been widely employed until now due to the increased control overhead and computational complexity at the routers. It is unlikely that IP multicast will be widely employed in the near future.

Broadband access to the Internet through services such as ADSL (asymmetric digital subscriber line) and FTTH (fiber to the home) has recently become very popular. Such services allow a large number of users to have their computers constantly connected to the Internet. Most computers are capable of storing many large files of digital content for the user, and many users are making such content available to others. Such technique is called peer-to-peer (P2P) networks. Each peer (i.e. user constantly connected to the Internet) can be a server to share its resources (supplying peer), while it is a client to obtain data from others (requesting peer). Due to this characteristic, application layer multicast (ALM) has recently proposed to implement the multicast functionality in application layer instead of IP layer, i.e. some of peers participating in the multicast play the role of the multicast router. Thus, ALM can be directly applicable to the current Internet since it does not need any new additional modification at IP routers. Application-level multicast techniques, such as NICE [1], SplitStream [2], SCRIBE [3], Narada [4], oStream [5] and Zigzag [12], construct the distribution trees over the peers to deliver video streams. However, it introduces another problem: it may overload some peers beyond their capacities. A peer in the tree may become a parent of several other peers. In the peer-to-peer networks, peers typically have limited capacity, especially of the upstream bandwidth. In many cases, peer cannot even provide the full stream rate to another peer.

In general, a P2P video streaming system has the following characteristics [7,13]: (1) peers have limited capacity, such as the restricted upstream bandwidth; (2) peers are heterogeneous in their bandwidth contribution. Therefore, each streaming session may involve multiple supplying peers. Xu [13] first proposed how to assign video data to multiple supplying peers in one streaming session. The goal is to ensure requesting peers can quickly initiate and then continuously playback a video, while it is being downloaded. However, Xu's work assumes the upstream bandwidth offered by a supplying peer must be one of the following values: $b/2$, $b/4$, $b/8$... $b/2^m$, b is the video consumption rate. This assumption is not flexible. To solve this problem, in this paper we propose a novel strategy to retrieve a long-duration video from multiple peers which have arbitrary and restricted upstream bandwidth, such that the waiting time for the requesting peer is minimized. In comparison with the Xu's work, our strategy can greatly improve the waiting time. For example the upstream bandwidth of supplying peers are $b/2$, $b/4$, $b/8$, and $b/8$, our strategy can improve the waiting time by 67%. To take into account the popular compressed video with variable bit rate, we also show how to apply readily to the VBR videos.

The rest of this paper is organized as follows. In Section 2, we review the Xu's work in detail. In Section 3, we present and analyze our strategy. How to apply our strategy readily to the VBR videos is also discussed in this section. In Section 4, its performance comparison is presented. Finally we make brief conclusions in Section 5.

2 Related Works

To the best of our knowledge, only the Xu's work [13] deals with problem which assign video data to multiple supplying peers in one streaming session such that the waiting time is minimized. Here, we first review some ideas used in it.

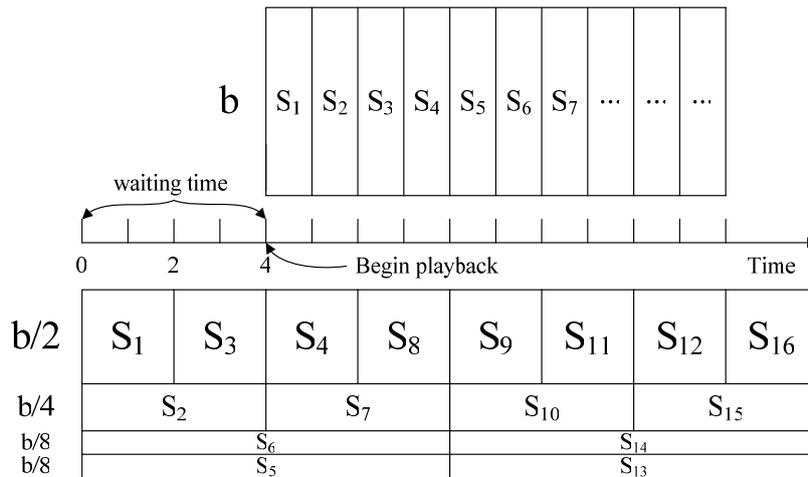


Fig. 1. Video data assignment of Xu's research.

Suppose the upstream bandwidth offered by a supplying peer must be one of the following values: $b/2, b/4, b/8, \dots, b/2^m, b$ is the video consumption rate and m is a positive integer. If there are n supplying peers which have been sorted in descending order according to their upstream bandwidth, and the lowest is $b/2^m$. Then it computes the assignment of the first 2^m segments, and the assignment repeats itself every 2^m segments for the rest of the video file. The segment assignment is in reverse order, such as $2^m, 2^m-1, \dots, 1$. Its assignment strategy is from top to bottom and from right to left. The minimum waiting time will be $n * \text{the length of a segment}$. An assignment example is shown in Figure 1. There are four supplying peers. The upstream bandwidth of supplying peers is $b/2, b/4, b/8$ and $b/8$; and the video playback time scale is described in top side.

3 Real-Time Video Streaming Data Assignment

In this section, we first present the problem more detail. Let the total length of the requested video be L , measured in seconds. Let the video consumption rate be b , measured in bits per second. Assume there are n supplying peers which the sum of upstream bandwidth is equal to the video consumption rate. The upstream bandwidth

of supplying peers are denoted as b_i , $i = 1, 2, 3, \dots, n$, measured in bits per second. The problem is: if we want to partition the whole video data into m segments (equal length is unnecessary), how to partition the video data and how to assign the segments to each supplying peer, such that the waiting time w is minimized. In other words, if we want to ensure the waiting time will be w , how to partition the video data and how to assign the segments to each supplying peer, such that the number of segments m is minimized. That is, the length of each segment must be as long as possible.

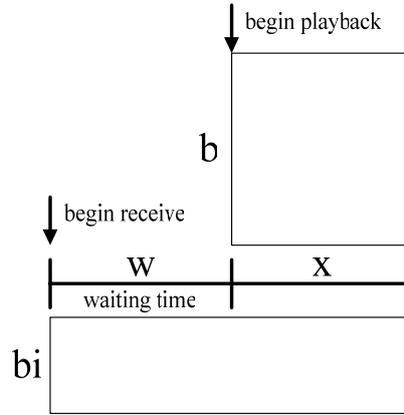


Fig. 2. The basic principle of our strategy.

3.1 Description of the basic principle

Now, we describe the basic principle of our strategy: Consider we want to retrieve video data from some supplying peer which upstream bandwidth is b_i , and the waiting time is w . Figure 2 shows this concept. From the figure, we observe that in order to guarantee continuous playback, the time x to retrieve the remaining portion must be not greater than the entire playback duration of the segment. In other words,

$$x \leq \frac{b_i(w+x)}{b} \quad (1)$$

Thus, by satisfying this condition (1), we ensure that the retrieval of the remaining portion will not affect the continuity of the playback at any time instant. Therefore the longest length of the segment will be $x = \frac{b_i}{b-b_i}w$, while

$$x = \frac{b_i(w+x)}{b} \Rightarrow bx = b_i(w+x) \quad (2)$$

3.2 The design and analysis of video retrieval strategy

In this section, we present our retrieval strategy in detail and determine the minimized waiting time w of requesting peer which retrieve m segments from all the n supplying peers, which have been sorted in descending order according to their upstream bandwidth.

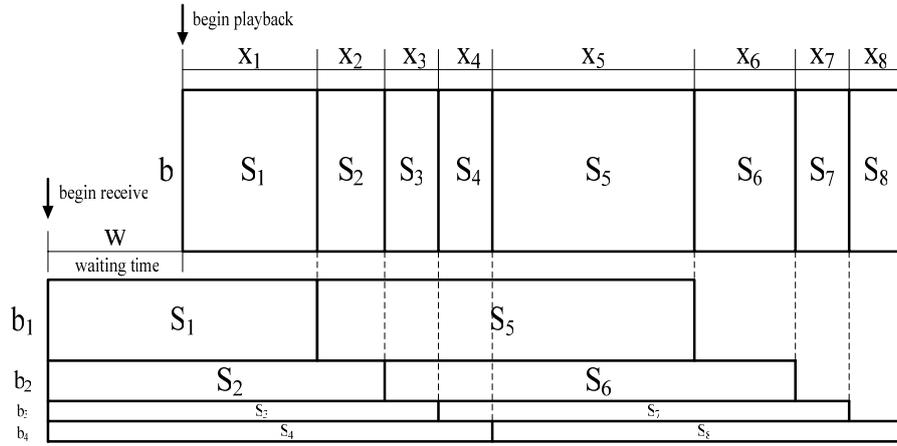


Fig. 3. An example of the video retrieval strategy.

According to the equation (2), if the upstream bandwidth of the supplying peer p_1 is b_1 and the waiting time is w , then the longest length x_1 of segment S_1 must satisfy the equation, $bx_1 = b_1(w + x_1)$. Now the waiting time for playback segment S_2 will equal to $w + x_1$. Hence the longest length x_2 of segment S_2 must satisfy the equation, $bx_2 = b_2(w + x_1 + x_2)$, while the upstream bandwidth of the supplying peer p_2 is b_2 . Figure 3 shows an example of the video retrieval strategy, while $m=8$ and $n=4$. Then we have the following recursive equations for the segments 1 to n .

$$\begin{cases} bx_1 = b_1(w + x_1) \\ bx_2 = b_2(w + x_1 + x_2) \\ bx_3 = b_3(w + x_1 + x_2 + x_3) \\ \bullet \\ \bullet \\ bx_n = b_n(w + x_1 + x_2 + \dots + x_n) \end{cases}$$

$$\Rightarrow \begin{cases} x_1 = \frac{b_1}{b-b_1} w \\ x_2 = \frac{b_2}{(b-b_1)(b-b_2)} wb \\ x_3 = \frac{b_3}{(b-b_1)(b-b_2)(b-b_3)} wb^2 \\ \bullet \\ \bullet \\ x_n = \frac{b_n}{\prod_{i=1}^n (b-b_i)} wb^{n-1} \end{cases}$$

If the number of the segment m is larger than the number of supplying peers n , then we can further have the following n recursive equations for the segments $n+1$ to $n+n$.

$$\begin{cases} bx_{n+1} = b_1(x_2 + x_3 + \dots + x_{n+1}) \\ bx_{n+2} = b_2(x_3 + x_4 + \dots + x_{n+2}) \\ \bullet \\ \bullet \\ bx_{n+n} = b_n(x_{n+1} + x_{n+2} + \dots + x_{n+n-1}) \end{cases}$$

$$\Rightarrow x_{n+i} = \frac{b_i}{b-b_i}(x_{i+1} + x_{i+2} + \dots + x_{i+n-1}), i=1 \sim n$$

For the segments $2n+1$ to $2n+n$, we can have the following n recursive equations.

$$\begin{cases} bx_{2n+1} = b_1(x_{n+2} + x_{n+3} + \dots + x_{n+n+1}) \\ bx_{2n+2} = b_2(x_{n+3} + x_{n+4} + \dots + x_{n+n+2}) \\ \bullet \\ \bullet \\ bx_{2n+n} = b_n(x_{n+n+1} + x_{n+n+2} + \dots + x_{n+n+n-1}) \end{cases}$$

$$\Rightarrow x_{2n+i} = \frac{b_i}{b-b_i}(x_{i+n+1} + x_{i+n+2} + \dots + x_{i+2n-1}), i=1 \sim n$$

And so on. Finally, we can induce the following recursive formulas (3) to determine the minimized waiting time w .

$$\begin{cases} x_p = \frac{b_p}{\prod_{i=1}^p (b-b_i)} w b^{p-1}, p \leq n \\ x_p = \frac{b_{[(p-1) \bmod n]+1}}{b-b_{[(p-1) \bmod n]+1}} (x_{p-n+1} + x_{p-n+2} + \dots + x_{p-1}), n < p \leq m \\ \sum_{i=1}^m x_i = L \end{cases} \quad (3)$$

3.3 Apply our strategy to VBR videos

Since the consumption rate of VBR videos usually varies with time. In order to guarantee continuous playback, we must initiate playback after finishing the retrieval of whole segment. Therefore, the size of segment S must be not greater than the size of retrieval data during the waiting time w . The equation (1) must be modified by following.

$$b_i w \geq \sum_{j \in S} f_j, \text{ where } f_i \text{ is the frame sequence, measured in bits} \quad (4)$$

Thus, by satisfying this condition (4), the longest length of the segment will be computed while

$$b_i w = \sum_{j \in S} f_j \quad (5)$$

Figure 4 shows an example of the video retrieval strategy, while $m=8$ and $n=4$. We can induce the following recursive formulas (6) to partition the video and assign to the supplying peers for a given waiting time w .

$$\begin{cases} b_p (w + \frac{\sum_{k=1}^{p-1} |S_k|}{F}) = \sum_{j \in S_p} f_j, p \leq n \\ b_{[(p-1) \bmod n]+1} \frac{\sum_{k=p-n}^{p-1} |S_k|}{F} = \sum_{j \in S_p} f_j, n < p \leq m \end{cases} \quad (6)$$

,where $|S_k|$ is the length of segment S_k , measured in frames; F is the video consumption rate, measured in frames per second.

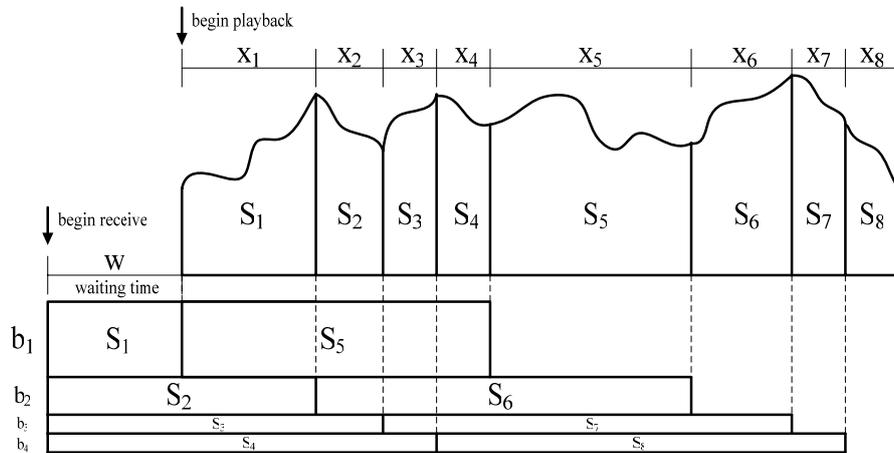


Fig. 4. An example of the VBR video retrieval strategy.

4 Performance Comparison

Review the goal is: if we want to partition the whole video data into m segments, how to partition the video data and how to assign the segments to each supplying peer, such that the access time w is minimized. Herein, we'll compare the viewers' waiting time with the Xu's work [13]. Assume the number of segments m is from 100 to 1000. Two arbitrary examples of upstream bandwidth of supplying peers are given as $b/2, b/4, b/8, b/8$ and $b/4, b/4, b/8, b/8, b/8, b/16, b/16$. Figure 5 show the first example, we can observe our strategy can greatly improve the viewers' waiting time by 67%. The second example is shown in Figure 6. Our strategy can still improve the viewers' waiting time by 59%.

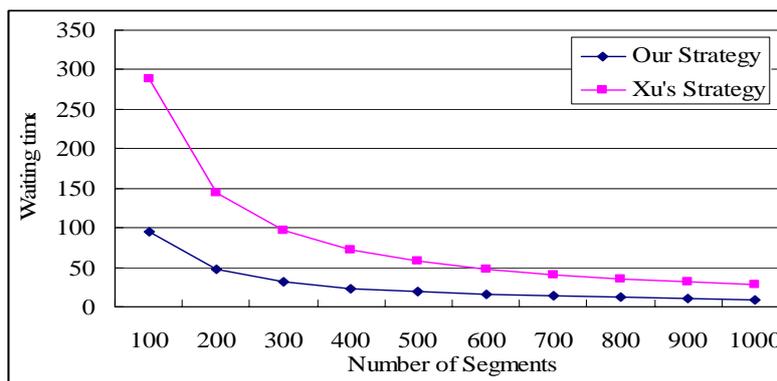


Fig. 5. Compare with viewers' waiting time with the Xu's work.

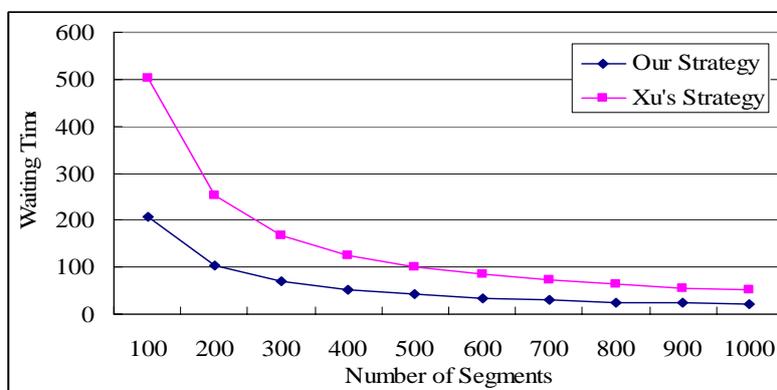


Fig. 6. Compare with viewers' waiting time with the Xu's work.

5 Conclusions

With the advancement of broadband networking technology, and the growth of processor speed and disk capacity, real-time video streaming service is getting increasingly popular among users and contributes a significant amount of today's Internet traffic. Recent research efforts have demonstrated the promising potential of building cost-effective video streaming systems on top of peer-to-peer (P2P) networks. Since the peers have limited capacity, such as upstream bandwidth, each streaming session may involve multiple supplying peers. In this paper, we propose a novel strategy to retrieve a long-duration video from multiple peers which have arbitrary and restricted upstream bandwidth, such that the waiting time is minimized. In comparison with the previous work [13], our strategy can greatly improve the waiting time. In the above-mentioned two arbitrary examples, our strategy can improve the waiting time by 67% and 59%. To take into account the popular compressed video with variable bit rate, we also show how to apply our strategy readily to the VBR videos.

References

1. S. Banerjee, B. Bhattacharjee, C. Kommareddy and G. Varghese, "Scalable Application Layer Multicast," *ACM SIGCOMM*, Aug. 2002.
2. M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron and A. Singh, "SplitStream: High-Bandwidth Content Distribution in a Cooperative Environment," *In Proc. of IPTPS*, Feb. 2003.
3. M. Castro, P. D ruschel, A. Kermarrec and A. Rowstron, "SCRIBE: A Large-Scale and Decentralized Application-Level Multicast Infrastructure," *IEEE Journal on Selected Areas in Communications*, Oct. 2002.

4. Y. Chu, S. Rao, S. Seshan and H. Zhang, "A Case for End System Multicast," *IEEE Journal on Selected Areas in Communications*, Oct. 2002.
5. Yi Cui, Baochun Li and Klara Nahrstedt, "oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, Jan. 2004.
6. Asit Dan, Dinkar Sitaram, Perwez Shahabuddin, "Dynamic batching policies for an on-demand video server," *Journal of Multimedia Systems*, vol. 4, no. 3, pp. 112–121, June 1996.
7. Mohamed M. Hefeeda, Bharat K. Bhargava and David K. Y. Yau, "A Hybrid Architecture for Cost-Effective On-Demand Media Streaming," *Journal of Computer Networks*, vol. 44, pp. 353-382, 2004.
8. L.S. Juhn and L.M. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Trans. on Broadcasting*, vol. 43, no. 3, pp. 268-271, Sep 1997.
9. Julian Liu, Su-Chiu Yang, Hsiang-Fu Yu, and Li-Ming Tseng, "Content Delivery Network with Hot-video broadcasting and Peer-to-peer Approach," *Journal of Information Science and Engineering*, Vol. 20, No. 6, 2004.
10. Anirban Mahanti, Derek L. Eager, Mary K. Vernon and David Sundaram-Stukel, "Scalable On-Demand Media Streaming with Packet Loss Recovery," *IEEE Trans. on Networking*, vol. 11, no. 2, pp. 195-209, April 2003.
11. Yu-Chee Tseng, Ming-Hour Yang and Chi-He Chang, "A Recursive Frequency-Splitting Scheme for Broadcasting Hot Videos in VOD Service", *IEEE Transactions on Communications*, vol. 50, issue:8, pp. 1348-1355, Aug 2002.
12. Duc A. Tran, Kien A. Hua and Tai T. Do, "A Peer-to-Peer Architecture for Media Streaming," *IEEE Journal on Selected Areas in Communications*, Jan. 2004.
13. Dongyan Xu, Mohamed Hefeeda, Susanne Hambrusch and Bharat Bhargava, "On Peer-to-Peer Media Streaming," *IEEE ICDCS*, July 2002.
14. Hung-Chang Yang, Hsiang-Fu Yu and Li-Ming Tseng, "Adaptive Live Broadcasting for Highly-Demand Videos," *Journal of Information Science and Engineering*, vol. 19, no. 3, pp. 531-549, May 2003.
15. Hsiang-Fu Yu, Hung-Chang Yang, Li-Ming Tseng and Yi-Ming Chen, "Simple VBR Staircase Broadcasting (SVSB)," *Journal of Computer Communications*, 2005.