PEER-TO-PEER VIDEO DELIVERY SCHEME FOR LARGE SCALE VIDEO-ON-DEMAND APPLICATIONS

Chen-Lung Chan*, Shih-Yu Huang, Mato Jan, and Jia-Shung Wang*
Department of Computer Science and Information Engineering, Ming Chuan University, Taiwan
*Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

ABSTRACT

On the basis of the emerging peer-to-peer structure and the characteristic of user’s behaviors, this paper proposes a scalable video delivery scheme, named P2PVD, for large scale video-on-demand application. In P2PVD, two types of orders are permitted: reserved and urgent orders. Reserved orders are encouraged with a lower price policy to smooth the network traffic and the server load. A delay-aware dynamic parallel transmission in the requesting peer is introduced to serve the reserved and urgent orders. The supplying peer is carefully designed based on three priority rules to increase the capacity of P2PVD. Experimental results indicate that P2PVD is scalable.

I. INTRODUCTION

Multimedia applications such as video-on-demand (VoD) allow users to transmit the video programs with VCR-like interactive operations over the networks. Scalability is an important issue to accelerate the marketability of VoD services. Simply speaking, a scalable video delivery scheme has the ability that the required channels of network and the load of server can be controlled in a limited range even when the number of clients or movies is extensively increased. However, most of the existing video delivery schemes such as batching [1-2] and patching [2-4] are not scalable, i.e., the number of channels is proportional to the number of clients when servicing a large number of videos. Another type of video service is the near VoD (NVoD) based on periodic broadcasting [5-6]. NVoD is a scalable scheme for video delivery, where the number of users is unlimited. However, it’s not scalable to the number of videos. For a two hour video program, 8 channels are required when the duration of periodically broadcast is set to be 15 minutes. Only a little number of videos can be serviced in NVoD because channels are limited resources.

The above phenomenon comes from two reasons. The first is that these schemes serve requests based on a real-time assumption trying to minimize the waiting time as little as possible. The second is that only a limited number of video servers can be used to provide services. Therefore, the server and the network commonly overload when the number of clients or movies is extensively increased. Actually, the above real-time assumption is not always correct because a lot of users usually rent a video tape, watch it when they have time, and return it to the place of rental. If user can order a video in advance, system can use the period of user plain-ahead time (UPAT) to download the ordered videos. Therefore, the network traffic and the server load can be smoothed because transmission of videos can be scheduled to avoid the peak traffic time. On the other hand, personal computer (PC) on the client has the ability to be a server now due to the progress on PC industry. Based on the above assumption, not only can each client act as a node retrieving videos from a server, it can be a proxy server for other clients. The more clients activated, the more proxy servers deployed. Accordingly, the load of video server can be reduced and the traffic is distributed in the network automatically. On the basis of the above characteristics, a peer-to-peer video delivery (P2PVD) scheme is proposed in this paper.

In fact, the peer-to-peer structure is widely utilized on applications of file sharing in recent years [7-9]. We usually exchange MP3 files on Napster [10] or Gnutella [11]. Researches on peer-to-peer video delivery are just emerging [12-13]. In [12], a peer-to-peer multimedia streaming and caching technique is presented to achieve low initial delay, small delay jitter, and small network utilization for real-time video applications. [13] constructs an application level multicast tree whose root is the original video server while peers are intermediate nodes and leaves. These architectures are effective when user requests are simultaneous and concentrated on a specific video stream such as in live-video streaming applications.

As stated, the feature of UPAT can also increase the scalability. This paper takes both advantages of UPAT and P2P to design platform for video applications. In the proposed P2PVD, two types of orders are permitted: reserved and urgent orders. Users are encouraged to make a reserved order in advance under a lower price policy. The ordered video would be transmitted to the storage of before the given deadline. In the contrast, users should pay a higher price when they make an urgent order. Instead of downloading the entire video program, the urgent order is serviced in a way of streaming. The P2PVD will service the urgent order with higher priority to satisfy the given timing constrain.

When a requesting peer orders a video, a set of supplying peers containing the desired video are first chosen to be servers. The requesting peer then dynamically parallel transmits (DPT) the video from the
suggested servers at the same time. The requesting peer first requests a different block of video from each supplying peer. When a block is transmitted by a supplying peer, the requesting peer initiates a new request for a block that has not yet been requested. The above process is repeated until all blocks are received. Therefore, fast supplying peers will deliver larger portions of a video and the traffic would be automatically shifted from congested parts to other parts of the network. In order to force urgent orders have higher priority to be serviced, next requests of the urgent order will be submitted immediately in DPT. Next requests of reserved orders would be delayed. The delay-aware DPT is the core to support reserved and urgent orders in P2PVD.

The rest of this paper is organized as follows. The main features of the proposed P2PVD scheme are described in Section II. Section III gives experimental results. Some conclusions are drawn in Section IV.

II. Peer-to-Peer Video Delivery Scheme (P2PVD)

Figure 1 is a simple demonstrative structure of P2PVD. The video programs partitioned into small blocks are stored in a video server S in the beginning. Each client initially acts as a requesting peer retrieving video from the server S. Once the client owns a video, it also acts as a supplying peer for other clients. By the above peer-to-peer scheme, the video contents are distributed in peers eventually. A manager M is employed to cooperate and control the overall processes among the video server and clients.

The video service works according to the order/transmit/view concept. When a client C_i wants to view a video program, it first orders the video from M. The format of the order message is given as follows.

\[ C_i \rightarrow M: \text{VIDEO\_ORDER}(V_i, D_i), \]

where \( V_i \) and \( D_i \) are the identification of the ordered video and the corresponding deadline of waiting time, respectively. \( D_i = 0 \) represents an urgent order.

In the manager M, a table \( T \) is utilized to record the information of cached videos in clients. Each entry \( (C_{\alpha}, V_{\alpha}) \) of the table \( T \) indicates that client \( C_{\alpha} \) can be a server providing the video \( V_{\alpha} \). When the manager receives an order message, the manager M utilizes the parameter \( D_i \) for billing and queries \( T \) by keyword \( V_i \) to check whether the ordered video cached in other peers or not. If there is no peer cached \( V_i \), the address of the original video server \( S \) should be responded to the requesting peer to inform \( C_i \) to get the video from \( S \). Otherwise, the response addresses include the server’s address and the addresses of the obtained supplying peers. Suppose there are two peers \( C_j \) and \( C_k \) already cache the ordered video, the response message is

\[ M \rightarrow C_i: \text{VIDEO\_RESPONSE}(S, C_j, C_k). \]

When the requesting peer \( C_i \) receives the response message, a delay-aware DPT method is utilized to get the ordered video. Suppose \( S, C_j \) and \( C_k \) are the suggested supplying peers from the manager, \( C_i \) first requests one block from \( S, C_j \) and \( C_k \). Every time a block has been received from a supplying peer, \( C_i \) will generate a next request to this supplying peer for another block that has not yet been requested. The above process would be repeated until all blocks are received. Finally, \( C_i \) reassembles all blocks to reconstruct the whole video. Details’ design of the requesting and supplying peers will be presented in the next subsection.

Once a requesting peer \( C_i \) completely receives the ordered video \( V_i \), the client should register its video information to the manager \( M \). The video registration message contains the following information.

\[ C_i \rightarrow M: \text{VIDEO\_REGISTRATION}(C_i, V_i). \]

When the manager receives the above video registration message, the pair of \( (C_i, V_i) \) is inserted into the table \( T \) to indicate that \( C_i \) can be a server for video \( V_i \) now. The above processes are summarized in Fig. 2.

A. Design of Requesting Peer

As mentioned, P2PVD provides reserved and urgent orders. In order to let urgent orders have higher priority to be serviced, the next request of urgent order will be submitted immediately and the new request of the reserved order would be delayed in the requesting peer. For an urgent order, the delayed time of the next request, \( T_d \), is set to be 0, i.e., the next request of the urgent order will be submitted immediately. For a reserved order, the traffic of the network and the corresponding deadline should be considered to design an efficient \( T_r \). If the current traffic is not heavy, the next request of the reserved order will be submitted as soon as possible, where the traffic of the network is approximately measured by the response time of the received block \( T_r \). Similarly, if the remaining time \( T_r = (D_r - T_r) \) is set to be \( T_r \) to the deadline \( D_r \) is small, the next request of the reserved order should also be submitted as soon as possible. In this paper, \( T_d \) is set to be \( T_r \) is larger than a given threshold \( \varepsilon \). Otherwise, \( T_d \) is set to be 0.

B. Design of Supplying Peer

Supplying peers can be viewed as proxy servers of requesting peers. It is possible that multiple peers simultaneously request a same supplying peer to deliver video. The capability of P2PVD is strongly affected by the serving order of requesting peers because a requesting peer becomes a supplying peer once it owns video. It’s useful to let a requesting peer with large transmitting bandwidth to be a supplying peer as soon as possible. Therefore, the following priority rules are the guideline to serve requesting peers in a supplying peer.

I. The priority of requesting peers with urgent orders is higher than that with reserved orders.

II. The requesting peers with larger transmitting bandwidth have higher priority.

III. The requesting peers with smaller distance to the supplying peer have higher priority.

Suppose the transmitting bandwidth of the
supplying peer to be $B$. A set of serving peers, $SP$, is first determined by the type of requesting peers. Based on the first rule stated above, $SP$ is set to be the set of requesting peers with urgent request if they exist. Otherwise, $SP$ is set to be the set of peers with reserved request. Let $SP = \{SP_1, .., SP_k\}$, where $k$ is the number of serving peers. A weighting factor, $W_i = \frac{B_i}{D_i}$, is introduced to assist the implementation of the second and third rules, where $B_i$ and $D_i$ are the corresponding bandwidth and distance of active peer, $SP$. Suppose $SP_g$ be the peer with largest weighting. The serving bandwidth, $B_g$, of each serving peer except $SP_g$ is defined as

$$B_g = B - \sum_{j \in SP-P(S)_g} B_j. \quad (1)$$

Then, the resting bandwidth of the supplying peer is assigned to $SP_g$, i.e.,

$$\overline{B}_g = B - \sum_{j \in SP-P(S)_g} B_j. \quad (2)$$

C. Timing Constrains on Reserved and Urgent Orders

Although P2PVD do it best to get the ordered and urgent videos, orders may be rejected when the network traffic is too heavy to support the deadline and real-time urgent videos, orders may be rejected when the network loading rate satisfy the real-time constrains for reserved and urgent orders, respectively.

In P2PVD, we force all urgent orders to wait a preset waiting time, 3 minutes in our design. The P2PVD preloads the video during the preset waiting period. To satisfy the real-time constrain of urgent orders, the loading rate $\alpha$ must greater than the playback rate $\beta$. When the above case occurs, the requesting peers is served in a way of streaming, i.e., the requesting peer starts to playback the video as the video data is still in transfer. Otherwise, the order is rejected.

For the reserved orders, P2PVD try to download the entire video to the requesting peer before the assigned deadline. However, the video wouldn’t be fully downloaded when the user wants to view the video. Let the amount of the current downloaded and unloaded video be $a$ and $b$, respectively. To let the requesting peer successfully watch the video, the video should be loaded before played, i.e.,

$$(a+\alpha \times \beta i) \geq \beta x i, 0 \leq i \leq b/\alpha. \quad (3)$$

where $b/\alpha$ is the time of the video completely downloaded. When $i = b/\alpha$, Eq. (4) can be derived as

$$a + \alpha \times \frac{b}{\alpha} \geq \beta \times \frac{b}{\alpha}. \quad (4)$$

Thus we have another inequality

$$\alpha \geq (\frac{b}{a+b}) \times \beta. \quad (5)$$

Therefore, a reserved order would also be rejected if

$$\alpha < (\frac{b}{a+b}) \times \beta. \quad (6)$$

III. EXPERIMENTAL RESULTS

This section studies performances of various VoD schemes measured by the reject rate. The simulated network is modeled by a 3-level tree structure. Each leaf node contains 1,000 clients and there are 8,000 clients in the network. The video server located at the root of the network contains 50 video programs whose length is fixed as 120 minutes and access pattern is according to a Zip-like distribution with skew factor 0.7. We assume that the video server has a capacity of transmitting 200 video streams simultaneously and the transmitting bandwidth of each client is randomly set as a value ranging from 1 to 10 video streams.

To show effects of peer-to-peer structure on video streaming, the original on-demand scheme, batching scheme with length 5 minutes, patching scheme with length 5 minutes, and the proposed P2PVD with urgent orders only are first compared. Figure 3 gives the corresponding reject rates for various inter-arrival times.

It’s easily seen that the proposed P2PVD is more scalable than the original schemes. Performances of original schemes are very poor. The reject rate is larger than 5% when $\lambda < 30$. In contrast, the reject rate of the proposed P2PVD is about 3% when $\lambda = 10$. When $\lambda$ further decreases to 6, reject rates of the original on-demand scheme, batching scheme, and patching scheme are 81.2%, 79.4% and 74.2%, respectively. These values are too high to be accepted. However, reject rate of P2PVD is only 7.8% under $\lambda = 6$ because the more clients activated, the more proxy servers deployed, the lower reject rate achieved.

Now, we analyze effects of the proposed weighted supplying peer mechanism on P2PVD (weighted P2PVD). For comparisons, the averaging mechanism is also implemented (averaging P2PVD). The corresponding reject rates are also shown in Fig. 3. Weighted P2PVD is especially powerful when the inter-arrival time is very small. Reject rate is down about 11% by weighted P2PVD over averaging P2PVD when $\lambda = 1$.

To exhibit effects of reserved orders, simulations are persisted for 12 hours in which $\lambda$ is initially set as 10, decreased to 1 in the third and fourth hours to simulate burst requests on hot times, and then increased the normal rate 10; one half of requests are set as urgent type and the others are reserved type with various UPAT. The corresponding results are given in Fig. 4. Obviously, the larger UPAT involved, the lower reject rate achieved because the burst traffic in the third and fourth hours can be smoothed to the later. Reject rates are always limited in 2% when UPAT is set as 240 minutes.
IV. CONCLUSIONS

Compared with most of existing technologies, such as on-demand VoD, batching and patching, the scalability feature of P2P structure makes it more attractive on video applications. In this paper, the scalability of P2P is further improved by taking the user plan-ahead time into account because many requests submitted ahead of time can provide a great opportunity for efficient scheduling requests, as a result, reducing the peak network traffic and server load. In order to rapidly increase the capacity of system, a delay-aware dynamic transmission and three priority rules are presented in requesting and supplying peers.

Acknowledgements
This work was partly supported by the National Science Council, R.O.C., under Grant NSC 92-2213-E-130-004.

REFERENCES