ENABLING LARGE-SCALE PEER-TO-PEER STORED VIDEO STREAMING SERVICE WITH QOS SUPPORT

Masaru Okuda, Murray State University

Abstract

The aim of this study was to enable large-scale, high-volume, stored-video streaming service over the peer-to-peer (P2P) network with Quality of Service (QoS) support. The primary focus of this paper is to address the following technical challenges associated with the distribution of stored streaming video through P2P networks: 1) allow peers with limited transmit bandwidth capacity to become contributing sources; 2) enable discovery of time-changing and time-bounded video frame availability at participating peers; and, 3) minimize the impact of distribution source losses during video playback.

To meet the above requirements, a new video distribution model was proposed which is a hybrid between client-server and P2P. In this model, the total length of a video is divided into a sequence of small segments. The peers execute a noble scheduling algorithm to determine the order, the timing and the rate of segment retrievals from other peers. The model employs an advertisement scheme that allows the discovery of video segment availability at other peers by incorporating parameters of the scheduling algorithm. An accompanying QoS scheme can reduce the number of video playback interruptions, while one or more video sources depart from the distribution network prematurely. The simulation study confirmed that the QoS scheme over the proposed distribution network was effective against excessive network delays, including multiple instances of distribution source losses.

Introduction

With the advent of recent technological advances, multimedia streaming service over the Internet is gaining increasing importance. Video streaming applications, such as high-definition video streaming and on-line DVD and Blu-ray rentals, have a potential to enrich our lives and create new business opportunities. As high performance end-user systems are becoming widely available and the number of subscribers to high-speed Internet access services is rapidly increasing, a new computing paradigm known as a Peer-to-Peer (P2P) network has emerged. P2P enables direct exchange of contents among a group of end users without the need for a centralized management structure. P2P offers a framework in which a large-scale, distributed and self-organizing content distribution network (CDN) can be constructed.

Although P2P has the potential to overcome the scalability problem associated with traditional client-server-based CDNs, it brings a set of new challenges. First, because uplink capacity of typical broadband access technologies is limited when compared to downlink, only a small subset of participating peers may be able to become contributing sources when bandwidth-intensive content is distributed. Second, since video streaming allows discarding of video frames anytime after their playback, the availability of video frames in user buffers for access by others becomes time-bounded and time-changing such that it aggravates the challenge associated with content discovery on P2P networks. Third, due to uncertainty in the behavior of peers sourcing video, users may experience excessive delays in video reception. As video streaming requires an orderly and timely delivery of video frames for a smooth playback, a video distribution scheme which minimizes the impact of distribution source losses on P2P networks is desired.

To achieve these goals, the authors proposed the design of a new video distribution network model and accompanying video segment discovery and reception schemes with QoS support. The following contributions were made through this research:

- Design of a new streaming video distribution network model called Virtual Theater Network.
- Design of a segmented video stream reception scheme and accompanying scheduling algorithm for orderly and timely video segment retrievals. It enables users with limited transmit bandwidth (i.e., transmit bandwidth << the nominal streaming rate) to become contributing sources.
- Design of an advertisement scheme for the discovery of available video segments in user buffers which incorporates the parameters of the video reception scheduling algorithm. It greatly simplifies the discovery process such that one advertisement and one query are sufficient to post and retrieve the lifetime video segment availability of a user.
- QoS support in mitigating the video viewing interruptions in the face of excessive delays, including ones caused by multiple video distribution source losses.

Related Work

Previous studies on P2P-based video distribution networks can be classified into two categories depending on the number of distribution sources from which participating users may request video streams. Under the single distribution source model, a requesting peer receives the entire video stream from one peer in the network. Depending on the schemes, peers self-organize themselves in a logical topology either in the form of chains [1], loops [2] or trees [3], [4]. The aim of these networks is to support live-media applications such as news tickers and real-time stock updates, which distribute low-bandwidth contents to many users. These schemes fall short in the support of high-bandwidth stored streaming video distribution services because they have no consideration for the asymmetric bandwidth availability of most of the widely used broadband access services available to consumers.

Under the multiple distribution source model, a requesting peer receives video streams from multiple peers. The combinations or concatenations of all streams reconstruct the original video. Some schemes split the video into multiple decodable layers, each of which is transmitted from a different source [5], [6]. While this approach avoids total loss of service in times of network failures and source losses, it incurs a large overhead to support the resiliency design. Other schemes divide the total length of the video in time and a sequence of video segments is transmitted from different sources [7-9]. The scheme presented here belongs to this category. The major difference between this scheme and those proposed by other authors lies in the assumption of how long the received video segments will remain in the user system once they are played back. This scheme assumes that they are discarded after a certain period of time and that their availability for retrieval by other peers is time bounded. Other schemes assume, implicitly or explicitly, long-term availability of downloaded content at the users' permanent storage system.

Architecture

Virtual Theater Network is a network model that aims to enable large-scale, on-demand, peer-to-peer stored-video streaming service over the Internet, which incorporates a hybrid architecture between client-server and peer-to-peer computing. Central to this model is a set of Virtual Theaters which provide a means to mass distribute video streams to communities of users. Within each Virtual Theater there exists a content distributor, known as a VT Distributor. A VT Distributor manages one or more VT Rooms in order to service the video distribution needs of users. A VT Room is a group of peers that forms a P2P community to receive and distribute a video stream.

Figure 1 illustrates the Virtual Theater Network model. There are multiple instances of Virtual Theaters throughout the Internet and this is depicted in the figure as Virtual Theaters 1, 2 and N. In this example, Virtual Theater 1 consists of VT Distributor 1 and three VT Rooms: VT Rooms 1, 2 and M. Each VT Room distributes a different video title and is created when the first user begins receiving the video feed from the VT Distributor. Subsequent users desiring to watch the same title of the video join the respective VT Rooms. As they join, they discover other peers in the room. Small circles within each VT Room in the figure represent the peers that joined the VT Room. The video in the VT Room is divided into a time sequence of small segments. The peers discover them in the buffers of other peers and retrieve them in their playback order. As video segments are being downloaded, the receiving peer makes them available for others to retrieve.

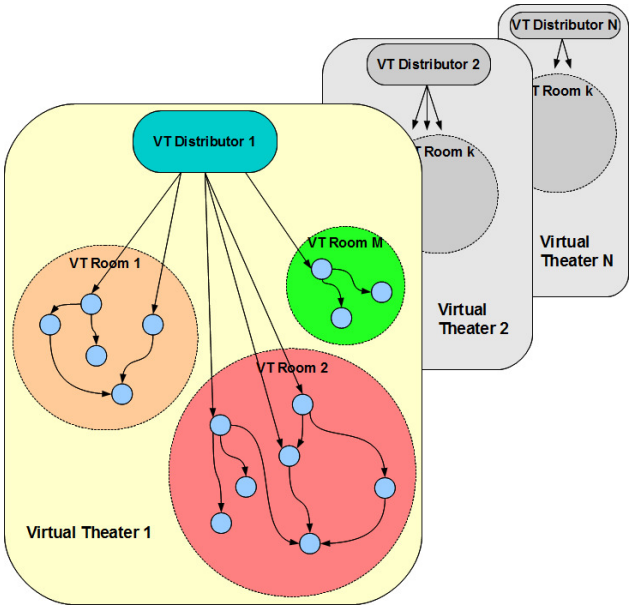


Figure 1. A Conceptual View of Virtual Theater Network

The main challenge in the design of video distribution service based on the proposed architecture is threefold: 1) how to manage the orderly and timely delivery of video segments that contribute to the self-sustainability of a VT Room, 2) how to organize dynamically changing video segment availability information within a VT Room to provide effective video segment advertisement and discovery service, and 3) how to mitigate the video viewing interruptions in face of excessive delays due to video distribution source losses. These issues are addressed in the following sections.

Video Segment Reception Management

This section describes Sliding Batch, a video segment reception scheme used in a VT Room and defines key concepts. The video segment reception scheme in Sliding Batch is expressed in terms of segments, epochs and batches. The definition of each is given below.

Segments

A video stream is a continuous flow of a sequence of compressed video frames transmitted over a network so that the recipient may play back the video frames as they arrive. In Sliding Batch, a block of a sequence of video frames makes up a Segment, S_i, and the concatenation of segments in their correct time sequence creates a video, V. Segments are similar in concept to chapters in a DVD and may vary in size and length. The number and size of segments in a video are VT-Room specific parameters. A segment is the basic unit of video exchanges among the users of a VT Room.

A segment, S_i, is characterized by its sequence position, i, in V, a set of frames, f_k, that belongs to S_i, its starting playback time, $\alpha(S_i)$, and the batch it belongs to, $\beta(e_j)$

$$V = \{S_i, i = 1,2,\cdots,N\}$$

$$S_i = \begin{cases} \{f_k, k = 1,2,\cdots,n_i, \quad n_i \leq F\} & if\ i = 1 \\ \{f_k, k = n_{i-1}+1, n_{i-1}+2,\cdots,n_i, \quad n_i \leq F\} & \\ \qquad if\ 2 \leq i \leq N & \end{cases}$$

$$S_i \cap S_j = \emptyset$$

$$\alpha(S_i) = \begin{cases} t_0 & if\ i = 1 \\ \alpha(S_{i-1}) + \delta(S_{i-1}) & if\ 2 \leq i \leq N \end{cases}$$

$$S_i \in \beta(e_j)$$

where N is the total number of segments in V, F is the last frame number in V, t_0 is the time the user joined the VT Room and began playing back the first segment, S_i, and $\delta(S_i)$ is the playback duration of S_i. The description of $\beta(e_j)$ is included later in this discussion.

Let $|V|$ and $|S_i|$ be the size of V and S_i, respectively. Let $\delta(V)$ be the total video playback time. Then, η, the nominal streaming rate of a video is given by $\eta = |V| / \delta(V)$, where $|V| = \sum_{i=1}^{N} |S_i|$. $\sum_{i=1}^{\delta}$ Accordingly, the playback duration of S_i, $\delta(S_i)$, is defined as $\delta(S_i) = |\delta(S_i)| / \eta$.

Epochs

In Sliding Batch, the lifetime of a video stream is divided into a sequence of time intervals, known as epochs. There are N epochs in a V and their duration may vary from epoch to epoch. Both the number and duration of epochs in a video are VT-Room-specific parameters. An epoch, e_i, is characterized by its starting epoch time, $\alpha(e_i)$, its duration, $\delta(e_i)$ and its associated batch, $\beta(e_i)$. An epoch is closely related to the playback property of a segment and described by Equations (1) □ (3).

$$\alpha(e_i) = \alpha(S_i) \tag{1}$$

$$\alpha(e_{i+1}) = \alpha(e_i) + \delta(e_i) \tag{2}$$

$$\delta(e_i) = \frac{|S_i|}{\eta} = \delta(S_i) \tag{3}$$

Batches

A batch, $\beta(e_i)$—each of which is associated with an epoch—is a set of segments whose downloads are initiated simultaneously at the beginning of (e_i). There are total of N batches in a video and each batch consists of a set of segments unique to itself, except for those batches with an empty set of segments. $\beta(e_i)$ is characterized by an associated epoch, e_i, a set of video segments, and a set of streaming sessions that are initiated at epoch e_i, each with rate r_j; refer to Equation (4).

$$\beta(e_i) = \begin{cases} \{S_j, j = 1,2,\cdots,n_j, \quad n_j \leq N\} & if\ i = 1 \\ \{S_j, j = n_{j-1}+1, n_{j-1}+2,\cdots,n_j, \quad n_j \leq N\} & \\ \qquad if\ 2 \leq i \leq N\ and\ n_{j-1} < N & \\ \emptyset \qquad otherwise & \end{cases} \tag{4}$$

$$\beta(e_i) \cap \beta(e_k) = \emptyset$$

$$A(S_j) = \alpha(e_i), \qquad \forall S_j, \qquad S_j \in \beta(e_i)$$

where $A(S_j)$ is the starting download time of S_j.

The ending download time of segment j, $\Omega(S_j)$, differs from segment to segment and is the ending playback time of segment j; refer to Equation (5).

$$\Omega(S_j) = A(S_j) + \frac{|S_j|}{r_j}, \quad S_j \in \beta(e_i) \tag{5}$$

Figure 2 illustrates the relationship between segments, epochs and batches in a simplified video reception scenario. In this example, a video is divided into four segments (N = 4) of varying lengths. Batch $\beta(e_1)$ consists of segments S_1 and S_2. Segment downloading for $\beta(e_1)$ was initiated at time

$\alpha(e_1)$ for both S_1 and S_2 at rates r_1 and r_2 , respectively. Batch $\beta(e_2)$ consists of segments S_3 and S_4. Segment downloading for $\beta(e_2)$ was initiated at time $\alpha(e_2)$ at rates r_3 and r_4 , respectively. No segment was associated with $\beta(e_3)$ or $\beta(e_4)$.

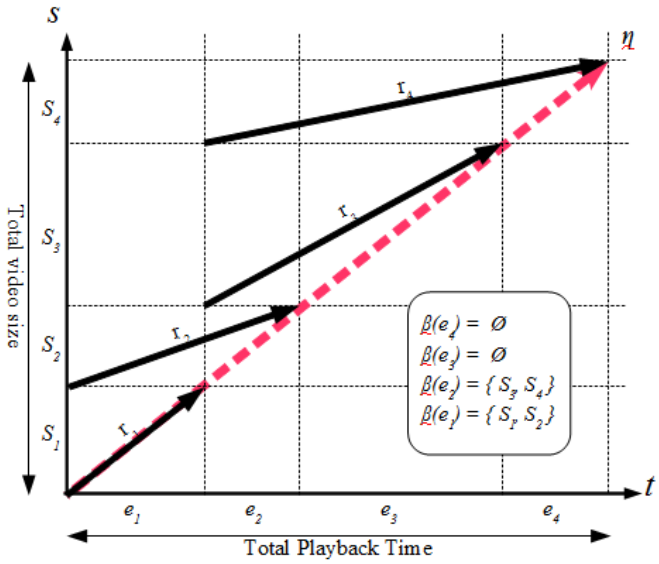


Figure 2. Relationship Among Segments, Epochs and Batches

Batch Size Determination

An important parameter of Sliding Batch is the number of segments in a batch, or the size of a batch, $|\beta(e_i)|$. It determines the beginning downloading time and the rate of each segment in the batch. The batch size, $|\beta(e_i)|$, is the total number of segments that a user begins downloading simultaneously at time $\alpha(e_i)$ and was determined by each user's available receive bandwidth and buffer space; refer to Equation (6)

$$|\beta(e_i)| = min\left(\overline{N(e_i)},\quad |\beta_R(e_i)|,\quad |\beta_B(e_i)|\right) \quad (6)$$

where $\overline{N(e_i)}$ is the number of remaining segments yet to be downloaded at time $\alpha(e_i)$, $|\beta_R(e_i)|$ is the rate-limited batch size at time $\alpha(e_i)$ and $|\beta_B(e_i)|$ is the buffer-limited batch size at time $\alpha(e_i)$.

The number of remaining segments, $\overline{N(e_i)}$, is defined in Equation (7).

$$\overline{N(e_i)} = \begin{cases} N & if\ i = 1 \\ N - \sum_{k=1}^{i-1} |\beta(e_k)| & if\ 2 \le i \le N \end{cases} \quad (7)$$

The rate-limited batch size, $|\beta_R(e_i)|$, refers to the size of a batch being computed solely on the available receive bandwidth, R_A, of the user and is determined by the maximum number of concurrent segment downloading sessions that can be sustained at time $\alpha(e_i)$; refer to Equation (8)

$$|\beta_R(e_i)| = \begin{cases} m_1, \\ \quad \exists max(m_1) | \sum_{k=1}^{m_1} r_k \le R_T,\ m_i \le N \\ \quad\quad if\ i = 1 \\ m_i - m_{i-1}, \\ \quad \exists max(m_i) | \sum_{k=m_{i-1}+1}^{m_i} r_k \le R_A(e_i), \\ \quad\quad m_i \le N \quad if\ 2 \le i \le N \end{cases} \quad (8)$$

where $R_A(e_i)$ is the receive bandwidth available at time $\alpha(e_i)$ and $R_T = R_A(e_1)$.

Similarly, the buffer-limited batch size, $|\beta_B(e_i)|$, was computed solely on the available buffer size, B_A, as if there were an infinite amount of receive bandwidth available. $|\beta_B(e_i)|$ is determined by the maximum number of concurrent segment downloads that can be sustained at time $\alpha(e_i)$; refer to Equation (9)

$$|\beta_B(e_i)| = \begin{cases} m_1, \\ \quad \exists max(m_1) | \sum_{k=1}^{m_1} |S_k| \le B_A(e_1), \\ \quad\quad m_1 \le N \quad if\ i = 1 \\ m_i - m_{i-1}, \\ \quad \exists max(m_i) | \sum_{k=m_{i-1}+1}^{m_i} |S_k| \le B_A(e_i), \\ \quad\quad m_i \le N \quad if\ 2 \le i \le N \end{cases} \quad (9)$$

where $B_A(e_i)$ is the available buffer size at time $\alpha(e_i)$.

The details of receive bandwidth and buffer management in relation to the scheduling algorithm are given in a study by Okuda and Znati [10] that includes a description of Restrained Sliding Batch, a variant of Sliding Batch that tames the aggressive segment pre-fetch behavior of the original scheme.

Figure 3 illustrates an example of how a user may receive segments in batches. In this example, a streamed video consists of 24 equally sized segments. The total receive bandwidth, R_T, of the user is twice the nominal streaming rate of the video segment. The total download buffer space, B_D, can accommodate a maximum of 10 simultaneous segment downloads. Fifteen batches are needed to initiate downloading of all segments. Notice that the first two batches, $\beta(e_1)$ and $\beta(e_2)$, are rate limited while the next eight batches, $\beta(e_3)$ through $\beta(e_{15})$, are buffer limited.

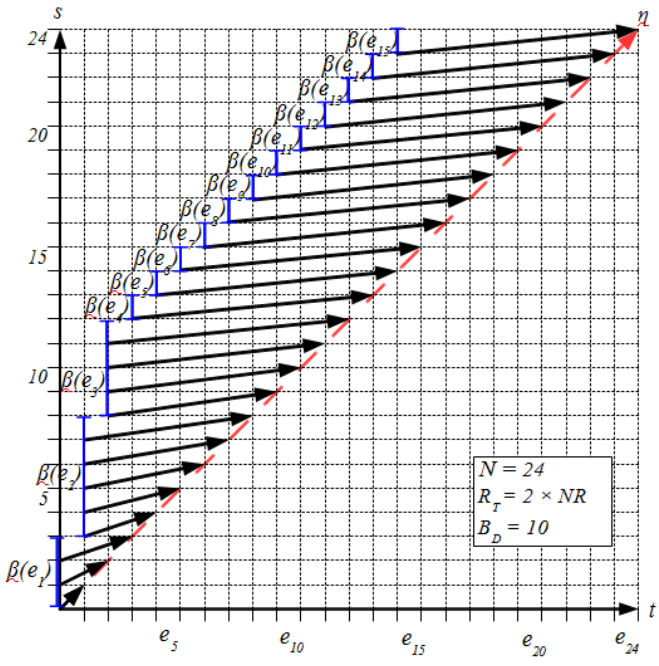


Figure 3. Sample Segment Receptions: Bandwidth and Buffer Limited Case

User Profile and VT Room Profile

The parameters used in Sliding Batch for the computation of batch size belong to either the VT Room profile or User profile. The VT Room profile describes the attributes of a video being distributed in a VT Room. These include parameters such as the size of a streaming video, $|V|$, playback duration, $\delta(V)$, the total number of segments, N, and the size of each segment, $|S_i|$. The VT Room profile is given to all users in each VT Room at the time they join the VT Distributor.

User profile describes the attributes of a user, primarily its resource availability, and consists of the following parameters: the time the user joined the VT Room, t_0, the total receive bandwidth set aside for the streaming service, R_T, the downloading buffer size, $|B_D|$, and the size of post-playback buffer space, $|B_H|$. $|B_H|$ determines how long a segment will remain in the user buffer after its playback. Users in a VT Room advertise their user profile through the advertisement and discovery scheme described in the next section.

Video Segment Advertisement and Discovery

This section describes Virtual Chaining, the video segment advertisement and discovery scheme used in a VT Room. Virtual Chaining allows users to cooperatively maintain a collection of user profiles, known as a state table, to share their segment reception state information with other users.

State Table

A state table is a collection of user profiles maintained cooperatively among the members of a VT Room. It describes each user's segment reception state and the transmit bandwidth availability. An entry in the state table consists of the following fields: IP address of the user advertising its state, parameters of the user profile (t_0, R_T, $|B_B|$, $|B_H|$), available transmit bandwidth (T_A), and the time of its entry. This is depicted in Figure 4.

IP	t_0	R_T	$\|B_D\|$	$\|B_H\|$	T_A	Time of Entry

Figure 4. Entry Fields of a State Table

An entry is added to the state table when a new user joins a VT Room. It is removed when the last video segment is dropped from the user's post-playback buffer. This condition can be determined by comparing the current time against when the user joined the VT Room plus the video playback duration and how long the video segments stay in the post-playback buffer; refer to Equation (10).

$$t_c > t_0 + \delta(V) + \frac{|B_H|}{\eta} \qquad (10)$$

State Table Sharing

The state table is shared among the users of a VT Room in the following manner. The VT Distributor maintains the tail-end portion of the state table, which contains user profiles of the last n users who joined the VT Room. A newly arrived user, U_i, receives the state table from the VT Distributor and reports its profile. The VT Distributor adds U_i's profile in the state table and drops the oldest entry if the table becomes greater than n. The VT Distributor waits for the next user arrival. In the meantime, U_i examines the received state table and tries to identify other users who may be able to provide segment distributions. If more users need to be discovered, U_i requests and maintains U_{i-n}, the oldest entry in the state table received from the VT Distributor. U_{i-n}'s state table consists of the user profiles of U_{i-n} through U_{i-2n} and potentially beyond if U_{i-n} had requested the state table from U_{i-2n}. This process is repeated until a qualified distribution source is located. If no qualified distribution source is found after stepping through the chain of state tables, U_i requests the direct video feed from the VT Distributor.

Figure 5 illustrates a sample trace of the state table sharing process. In this figure, circles represent users and shaded boxes below each circle represent portions of the state table maintained by each user. While all users maintain a portion of an overlapping state table, this figure only shows the ones maintained by users whose user ID is a multiple of n. Each shaded box contains n entries of user profiles. The user pointed to by the oldest entry in the state table is denoted by the dashed arrow extending from the box to the appropriate user. Solid arrows represent the transfer of state table entries.

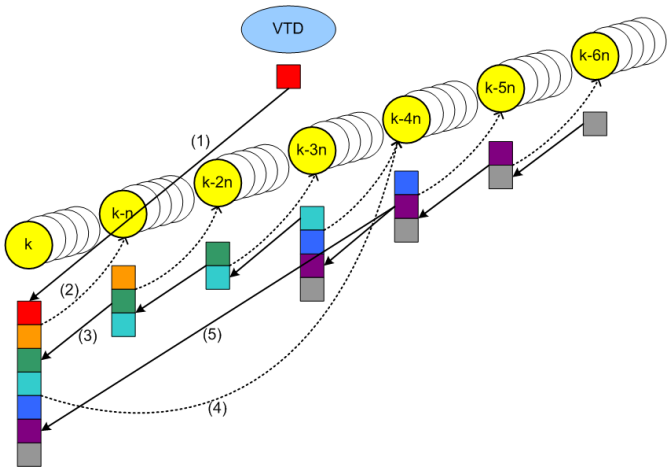


Figure 5. A Sample View of State Table Sharing Instances

A newly arrived user, U_k, joins a VT Room and receives the state table from the VT Distributor (step 1), which contains the user profiles of U_{k-1} through U_{k-n}. To discover more users in the VT Room, U_k requests older state table entries from U_{k-n} (step 2). U_{k-n} maintains the user profiles of U_{k-n-1} through U_{k-2n} it received from the VT Distributor at the time it joined. In this example, U_{k-n} also has the user profiles of U_{k-2n-1} through U_{k-4n}, which was received from U_{k-2n}. All of these entries are sent from U_{k-n} to U_k (step 3). To further discover older users, U_k requests U_{k-4n} to send its portion of the state table (step 4). U_{k-4n} sends the user profiles of U_{k-4n-1} through U_{k-7n} to U_k (step 5).

If U_k does not receive a response from the user (e.g., U_{k-n}) from which it requested an older state table, the next oldest entry in the state table (i.e., U_{k-n+1}) will be contacted.

Due to its simple operation, Virtual Chaining is relatively easy to implement, deploy, and study its behavior. A distributed and redundant state table, available at participating users, offers resiliency such that a loss of a few users does not break the segment advertisement, discovery or distribution operation. Virtual Chaining is fair, in terms of the carried workload among the users, such that no single user is expected to perform more work than any other. Virtual Chaining is also scalable in that the workload placed upon each user remains a constant regardless of the size of the membership in the P2P community.

QoS Support

Streaming applications that operate over a network with fluctuating traffic delays, such as the Internet, employ a playout buffer, B_P. The goal of the playout buffer is to prevent video frame starvation (i.e., absence of video frames in a buffer) during playback. This is achieved by pre-fetching an initial portion of a video stream and withholding its playback for a predetermined duration of time. The delay incurred by this operation is referred to as playout delay, D_P, or start-up delay. In exchange for inducing delay, it is hoped that the subsequent delays during the lifetime of video playback may be absorbed by the playout buffer. Playout delay should be long enough to cope with typical delays seen on the Internet, yet short enough for users to tolerate the initial waiting time.

The challenge to incorporating the traditional network delay coping mechanism in Sliding Batch is how to deal with the loss of distribution sources. To address this issue, a set of delay management mechanisms was proposed. Extended Playout Delay (EPD) and Expedited Segment Reception (ESR) work together to prevent buffer under-run conditions during multiple instances of distribution source losses.

Extended Playout Delay

Extended Playout Delay (EPD) is designed to prevent a buffer starvation condition after an excessive delay is detected. This is achieved by having users wait extra time before the initial video playback can begin. It differs from the traditional delay coping mechanism in that EPD introduces the concept of level of protection (n), which aims to separate the excessive delay detection period, (D_P), from the initial playout delay period (i.e., extended playout delay, D_E). Equation (11) depicts the relationship between n, D_P and D_E

$$D_E = n \cdot D_P = \frac{n \cdot |B_P|}{\eta} \tag{11}$$

where $|B_P|$ is the size of playout buffer. n corresponds to the number of times a user may encounter excessive delays (i.e., the number of distribution source losses) and not experience video presentation interruptions.

Figure 6 displays a logical view of a sample extended playout buffer, B_E, with an excessive delay protection level of $n = 4$.

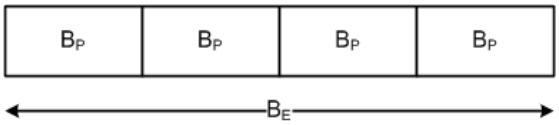


Figure 6. A Sample Extended Playout Buffer

Let D_{En} be the start-up latency introduced by EPD to provide an n^{th} level of excessive delay protection. At D_{E1} = 1 x D_P, EPD only provides delay absorptions up to D_P with no protection against a distribution source loss, just as the traditional playout buffer does. At D_{E2} = 2 x D_P, EPD offers a one-time distribution source loss protection during the lifetime of a segment download, in addition to delay absorptions up to D_P. It provides an uninterrupted video presentation if a user loses one distribution source and begins receiving the segment from a new distribution source. However, if the user loses the new distribution source, there will be a playback interruption. At D_{E3} = 3 x D_P, a two-time distribution source losses can be tolerated during the lifetime of a segment download, in addition to delay absorptions up to D_P. By extending the start-up delay, the level of protection against multiple instances of distribution source losses over the lifetime of a segment download can be improved.

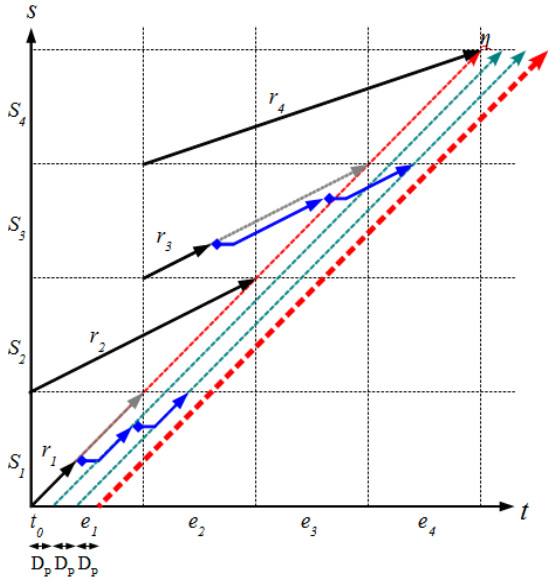


Figure 7. Segment Receptions with Extended Playout Delay

A sample segment reception with EPD is depicted in Figure 7. In this example, start-up delay has been extended to D_{E3} = 3 x D_P. The dotted arrow lines running diagonally across the middle of the figure represent the nominal streaming rates and show the playback positions of the streamed video at three levels of protections. Segments S_1 and S_3 both experience an excessive delay twice during their download, which is depicted by horizontal lines with diamond-shaped starting points. Once the excessive delay condition is declared, the user begins retrieving the affected segment from another distribution source. Note that, even after moving to a new distribution source twice, a sufficient amount of data has been pre-fetched in the extended playout buffer to provide a delay absorption up to D_P.

Expedited Segment Reception

Expedited Segment Reception (ESR) offers protection against a distribution source loss by increasing the segment downloading rate. By expediting the ending time of a segment download, ESR attempts to gain sufficient time to recover from delays experienced during segment receptions.

Let r_i^{En} be the rate of ESR for downloading segment S_i at n^{th} level of protection against excessive delays. At r_i^{E1} = $|S_i| / (\Delta(S_i) - D_P)$, given $\Delta(S_i) > D_P$, where $\Delta(S_i)$ is the downloading duration of S_i (i.e., $\Delta(S_i) = \Omega(S_i) - A(S_i)$), one-time distribution source loss protection can be achieved during the lifetime of an S_i download. It assures an uninterrupted video viewing experience by the user even if a distribution source is lost and the delayed segment is retrieved from a new distribution source. However, if the user loses the distribution source again, there will be a playback interruption. At r_i^{E2} = $|S_i| / (\Delta(S_i) - 2D_P)$, given $\Delta(S_i) > 2D_P$, two-time distribution source loss protection can be offered during the lifetime of an S_i download. At r_i^{E3} = $|S_i| / (\Delta(S_i) - 3D_P)$, given $\Delta(S_i) > 3D_P$, a three-time distribution source loss protection can be offered during the lifetime of S_i download. A higher degree of protection can be achieved with a relatively small amount of increase in the rate of segment download.

Figure 8 shows an example of how video segments may be received under ESR. In this example, S_1 is protected against one-time distribution source loss by receiving the segment at r_1^{E1}. S_2 and S_3 are protected against two-time source losses by increasing the reception rate to r_i^{E2}. At r_i^{E3}, S_4 can withstand three-time distribution source losses. The example shows that excessive delays have been observed while downloading segments S_1 and S_3, but they did not cause video playback interruptions because a sufficient amount of data was pre-fetched through ESR.

Downloading of each segment can be associated with a different degree of protection through ESR. For example, if a measurement shows that a significantly higher rate of distribution source loss is experienced in downloading S_i, a higher level of protection can be afforded to the reception of S_i. Let L_i be the maximum number of distribution source losses a user anticipates when downloading an S_i segment.

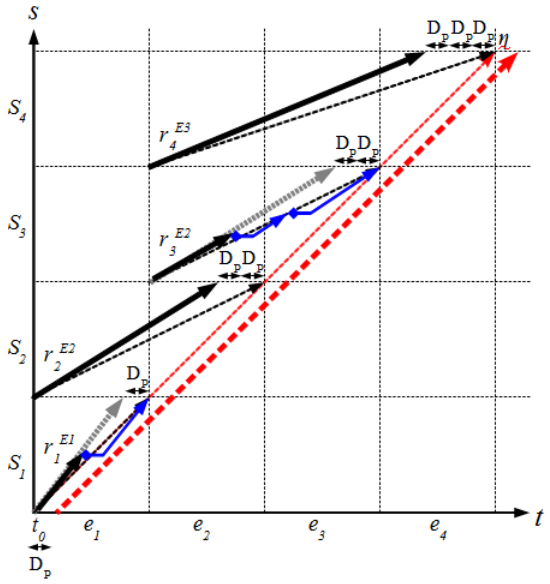


Figure 8. Sample Segment Receptions with Expedited Segment Reception

The expedited rate of reception, r_i^{EL}, that protects against L number of distribution losses is given by Equation (12).

$$r_i^{EL} = \frac{|S_i|}{\Delta(S_i) - L_i \times D_P} \qquad (12)$$

In order to determine the level of protection needed for each segment download, the statistic on the loss of distribution sources must be collected. This is achieved by informing the VT Distributor every time a user experiences a distribution source loss. The VT Distributor keeps the statistic of distribution source losses for each segment and shares it with users as they join the VT Room.

Let P_i be the probability of a user experiencing a distribution source loss when downloading S_i, assuming loss of distribution sources are IID. To achieve a successful download of S_i at or above a protection goal, g such as $g = 0.95$, the following condition must be met:

$$g \geq 1 - P_i^{L_i}$$

The level of protection needed for an S_i download can be computed by solving for L_i.

The user executes the Excessive Delay Protection Algorithm, as depicted in Table 1, to determine the amount of Extended Playout Delay and the rate of Expedited Segment Reception. The strategy used in this algorithm is to let the user wait as long as it is willing at the initial playback time through EPD. If necessary, expedite individual segment receptions through ESR can be expedited. Let L be the maximum level of protection required to meet g. Initially, L is set to the maximum value of L_i, the greatest level of protection required among all segment receptions. Let w be the maximum time a user is willing to wait for the start-up latency. If the extended playout delay, $D_E = L \times D_P$, is greater than w, L is set to $[w/D_P]$. This is the level of protection offered by EPD. For each segment that belongs to a batch, $\beta(e_i)$, a need for an additional level of protection through ESR is investigated. If the level of protection, L_i, required for downloading S_i is greater than the level of protection provided by the EPD (L), the rate at which the segment is downloaded will be increased to $r_i^E = |S_i| / (\Delta(S_i) - (L_i - L) \times D_P)$. If not enough receive bandwidth is available, the segment download will not be initiated.

Table 1. Excessive Delay Protection Algorithm

```
1:  // initialize
2:  L = max(Li)
3:  // let user wait as long as it is willing
4:  if (L x DP > w) then
5:  L = [w/DP]
6:  endif

7:  for each segment ∈ β(ei) d ∈ o
8:  // increase the download rate as needed
9:  if (Li > L) then
10: ri^E = |Si| / (Δ(Si) – (Li – L) x DP)
11: endif
12: // not enough RxBW to meet the protection
     requirement
13: if (ri^E > RA) then
14: print warning and break
15: endif
16: endif
```

Simulation Design and Analysis

This section describes the design and analysis of experiments performed on a Virtual Theater Network. A software model was created to simulate the behavior of a VT Room. Two types of experiment was conducted. The focus of the first type of experiments is to study how well the proposed video distribution scheme would alleviate the load on the VT Distributor under different operating environments. The results of the first set of simulation studies are available from Okuda and Znati [10]. The second set of experiments focuses on how well the proposed QoS scheme would mitigate the impact of distribution source losses on the video presentation. The description of the model, the design of the experiments, and the analysis of simulation study are given below.

Description of the Model

The simulated VT Room consists of a VT Distributor and a series of user processes that arrive at the VT Room. The VT Distributor supplies the parameters of the VT Room profile to the newly joining users, such as the total video playback time (120 minutes), the nominal streaming rate (1.0 Mbps) and the total number of segments (24) in the video. Note that the values in parentheses denote the default values used in the experiments. For simplicity of simulation for this study, the video was divided into equal segment lengths and equal playback times (5 minutes).

The user processes simulate the behavior of peers joining the VT Room, discovering other users, identifying possible distribution sources, receiving video segments, distributing video segments as requests arrive, and departing from the VT Room. The inter-arrival time of user processes is exponentially distributed (a mean of 10 seconds). To reflect the asymmetrical nature of the transmit and receive bandwidth capacity of typical broadband access technologies, each user was equipped with a fixed receive bandwidth (2.0 Mbps) and varying transmit bandwidth (30% to 100% of the receive bandwidth; uniformly distributed). Each user executes Virtual Chaining to identify possible distribution sources and implements Restrained Sliding Batch to receive video segments. All experiments simulate the bandwidth-limited network environment where a sufficient amount of download buffer exists at each user ($|B_H| \geq N$). The default post-playback buffer size allows a segment to remain in buffer for a finite period of time (15 minutes) after its playback.

Experimental Design and Analysis

Two sets of experiments were designed to study the effectiveness of QoS schemes in mitigating video presentation interruptions when users experience excessive delays while receiving video segments. Each experiment was measured against Chaining [1].

The first set of experiments studied the effects of the size of the extended playout delay in reducing the video presentation interruptions under different rates of premature user departures from the VT Room. Extended Playout Delays were varied from 1 x D_P to 4 x D_P. The probability of premature node departure, P, was varied from 0.1 to 0.4. The time a node may spend before prematurely departing from the VT Room was uniformly distributed during the playback of the entire video. The total number of video presentation interruptions experienced by participating users was normalized to the total number of distribution source losses being detected in the VT Room.

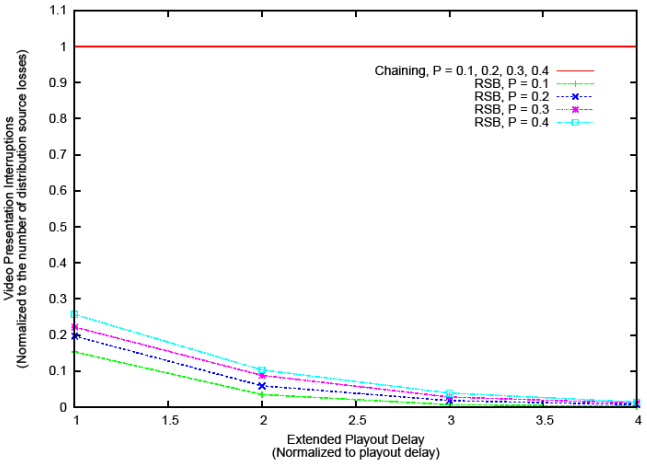


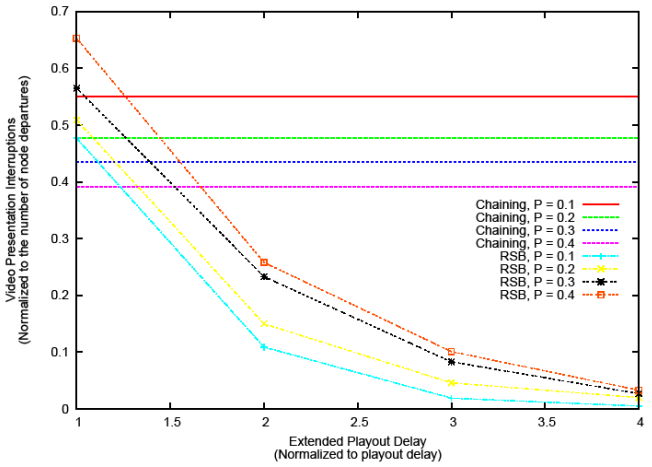
Figure 9. Effects of Extended Playout Delay □ I

In Chaining, all distribution source losses being detected by the users resulted in the video presentation interruptions, as they use a traditional playout buffer mechanism, regardless of the rate of user departures from the VT Room. This is depicted in Figure 9 by the horizontal line drawn at the 1.0 mark. In Restrained Sliding Batch, through the implementation of the proposed QoS scheme, not all distribution source losses being detected result in the interruption of the video presentation. The user may experience interruptions only if a segment reception encounters a greater number of distribution source losses than all other segments in the same batch (i.e., maximum distribution source losses of a batch, $L_{\beta(ei)}$) Furthermore, a video presentation interruption can occur only if the sum of the maximum distribution source losses of all batches results in an accumulated delay beyond the extended playout delay. Let Γ be the total number of video presentation interruptions being experienced by a user for the duration of the video playback. Γ is defined as:

$$\Gamma = \left\lceil \frac{\sum_i^N L_{\beta(ei)} \times \text{playout delay}}{\text{extended playout delay}} \right\rceil$$

In Restrained Sliding Batch, at 1 x D_P, 15% to 26% of distribution source losses being detected resulted in actual video presentation interruptions, when 10% to 40% of the nodes prematurely departed from the VT Room. At the extended playout delay of 2 x D_P, the rate of video presentation interruptions decreased between 4% and 10% when P was varied from 0.1 to 0.4. When the size of the extended playout buffer was increased to 4 x D_P, less than 1% of all distribution source losses being detected by users resulted in actual video presentation interruptions when 10% of the total nodes prematurely leave the VT Room. At a rate of 40% of premature node departure, roughly 1% of the detected source losses resulted in video presentation interruptions.

Figure 10. Effects of Extended Playout Delay □ II



Another way to express the effects of extended playout delay on the video presentation interruptions under different node departure rates is shown in Figure 10. In this figure, the total number of video presentation interruptions is normalized to the total number of premature node departures under the same settings, as shown in the previous set of experiments. Note, in Restrained Sliding Batch, a premature user departure may result in multiple instances of distribution source losses experienced by other users. Figure 10 shows how many instances of video presentation interruptions are introduced when one node departs prematurely from the service. In Restrained Sliding Batch, every premature node departure resulted in a video presentation interruption at the rate of 48% to 65% at 1 x D_P, when P is varied from 0.1 to 0.4. When the size of the extended playout buffer was doubled, the interruption rates halved at P = 0.4 and quartered at P = 0.1. When the extended playout delay was at 4 x D_P, less than 1% of premature node departure resulted in a video presentation interruption when P = 0.1 and roughly 3% of premature node departures resulted in video interruptions when 40% of users left the service prematurely.

In Chaining, every premature node departure resulted in a video presentation interruption at rates between 55% and 39% when P was varied from 0.1 to 0.4. At first glance, the simulation result seems counter intuitive in that the rate of video interruptions decreased as the rate of node departures increased. This is because, in Chaining, as the rate of node departure increases, a large percentage of users begin relying on the central server for video distribution feeds rather than their peers. The next set of experiments proves this point.

The second set of experiments studied the effects of premature node departure rates on the load on the VT Distributor under varied user arrival rates to the VT Room. The results are shown in Figure 11. The peak transmit bandwidth demand on the VT Distributor was normalized to the peak transmit bandwidth demand on a traditional client-server-based video distribution network. At P = 0, no node prematurely departs from the VT Room and the results from this simulation were used as reference. For $0.1 \leq P \leq 0.4$, both Chaining and Restrained Sliding Batch have similar rates of load increase on the VT Distributor as the rate of premature node departure increased. The main difference between the two schemes is that Restrained Sliding Batch with QoS extension requires only a third or less of resources from the central server when compared to Chaining.

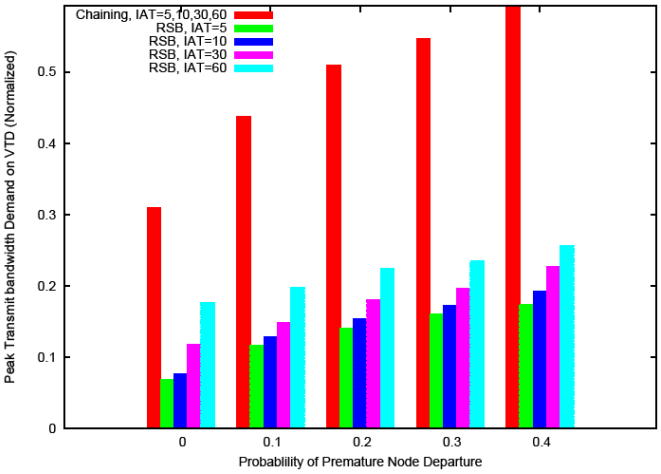


Figure 11. Effects of Node Departure Rate

These two sets of experiments verified the effectiveness of QoS schemes in reducing the number of video presentation interruptions, when users depart from the network prematurely. A linear increase in the size of the playout buffer resulted in logarithmic decreases in the rate of video presentation interruptions. Furthermore, relatively small increases in the VT Distributor load was observed when the probability of premature node departure was raised.

Conclusion

The authors proposed a design for a new streaming video distribution network model called Virtual Theater Network, which allows organization of peer-to-peer communities to support the distribution of videos among the community members. The model employs a segmented video stream reception scheme with its accompanying scheduling algorithm for orderly and timely video segment retrievals that allow contributions from users with limited transmit band-

width availability. QoS extension of the distribution scheme allows reduction in the number of video presentation interruptions when excessive delays are observed. The model also employs a video segment availability advertisement and discovery scheme, which incorporates the parameters of the scheduling algorithm. It enabled the advertisement and query of dynamically changing segment availability information of each user in one advertisement and one query. The simulation study showed dramatic improvements in the video presentation interruption occurrences under the proposed QoS scheme when distribution sources depart prematurely from the network.

References

[1] Sheu, S. & Hua, K. A. (1997). Virtual Batching: A New Scheduling Technique for Video-on-Demand Servers. *Proceeding of the International Conference on Database Systems for Advanced Applications*, (pp. 481-490).

[2] Kusmierek, E., Dong, Y. & Du, D. (2005). Loopback: Exploiting Collaborative Caches for Large-Scale Streaming. *Proceedings of ACM/SPIE MMCN*.

[3] Banerjee, S., Bhattacharjee, B. & Kommareddy, C. (2002). Scalable application layer multicast. *Tech. Rep., UMIACS TR-2002*.

[4] Tran, D., Hua, K. & Do, T. (2003). Zigzag: An efficient peer-to-peer scheme for media streaming. *Proceedings of IEEE INFOCOM*.

[5] Padmanabhan, V., Wang, H., Chou, P. & Sripanidkulchai, K. (2002). Distributing streaming media content using cooperative networking. *Proceedings of ACM/IEEE NOSSDAV*.

[6] Rejaie, R. & Ortega, A. (2003). PALS: Peer-to-Peer Adaptive Layered Streaming. *Proceedings of ACM NOSSDAV*.

[7] Hefeeda, M, Bhargava, B. K. & Yau, D. K. (2004). A hybrid architecture for cost-effective on-demand media streaming. *IEEE Computer Networks*, 44(3).

[8] Shan, Y. & Kalyanaraman, S. (2003). Hybrid video downloading/streaming over peer-to-peer networks. *Proceedings of IEEE ICME*.

[9] Hefeeda, M., Habib, A., Botev, B., Xu, D. & Bhargava, B. (2003). PROMISE: Peer-to-Peer Media Streaming Using CollectCast. *Proceedings of ACM Multimedia*.

[10] Okuda, M. & Znati, T. (2006). Virtual Theater Network: Enabling Large-Scale Peer-to-Peer Streaming Service. *Proceedings of Distributed Multimedia Systems*.

Biography

MASARU OKUDA received the B.S. degree in Information System and Computer Science from Brigham Young University - Hawaii, Laie, HI, in 1989, and the M.S. degree in Telecommunications and the Ph.D. degree in Information Sciences from the University of Pittsburgh, Pittsburgh, PA, in 1996 and 2006, respectively. Currently, he is an assistant professor of Telecommunications Systems Management at Murray State University, Murray, KY. His teaching and research areas include computer and network security, US telecom policies, network protocol analysis, network architecture design, QoS enabled networks, peer-to-peer networks, and video distribution networks. Dr. Okuda may be reached at masaru.okuda@murraystate.edu