Optimizations of Stored VBR Video Transmission on CBR Channel

Ray-I Chang, Meng Chang Chen, Jan-Ming Ho and Ming-Tat Ko
Institute of Information Science, Academia Sinica, Taiwan, ROC
{william,mcc,hoho,mtko}@iis.sinica.edu.tw

Abstract

In this paper, a new method is proposed to optimize stored VBR (variable-bit-rate) video transmission on CBR (constant-bit-rate) channel. The proposed method can minimize both the buffer requirement and work-ahead for a given peak transmission rate. Besides, the network utilization is maximized with the minimum service connection time. These problem parameters are not optimized at the same time in the conventional approaches. In this paper, we at first present the *Lazy* scheme to determine the minimum buffer and work-ahead required for a given peak transmission rate. Then, the *Aggressive* scheme is applied to maximize the system resource utilization (e.g., network bandwidth). The proposed schemes can be easily extended to transmit the VBR video with the minimum rate variability, or to resolve the buffer-constrained transmission problem with the minimum peak transmission rate for a given buffer size. Experiments to many well-known benchmark video traces show that the proposed method can obtain better network utilization and requires less memory buffer than the conventional approaches.

1 Introduction

The issues of distributed multimedia applications such as video conferencing, digital library, home shopping, and distance learning, has been the focuses to many researches of industry and academy society recently. Both live and stored audio and video media are transmitted over high-speed networks in some of these applications, and played back continuously at remote clients. For most applications, end-to-end quality-of-service (QoS) guarantees are required. There are two major network service types providing QoS guarantees in modern high-speed network technology (e.g., ATM), i.e., variable bit rate (VBR) and constant bit rate (CBR) services. As resources are usually allocated exclusively in fixed-size chunks to each service stream, it is relatively simple for networks to support CBR service [4] in terms of management complexity and overhead. However, digital media streams (e.g., audio and video) are usually VBR in nature due to compression

technology [3, 4]. It would be a waste of network bandwidth if peak rate allocation scheme is used to transmit a VBR stream because the average rate of some VBR stream (e.g., an MPEG video) is usually no more than 25% of its peak rate. One remedy to this problem is to use statistical service model that VBR streams are multiplexed to share the entire link bandwidth. The advantage of this model is that statistical analysis can theoretically guarantees a QoS bound with substantial less resources allocated to service streams than the peak rate model. However, on the other hand, admission control and network transmission scheduling at intermediate network nodes (e.g., output ports of ATM switches) are usually quite complicated in both analysis and implementation for the VBR service model. It is thus the aim of this paper to seek for a solution which transmits VBR stored video traffic with comparable or even better performance than existing ones.

The performance of a transmission scheme is usually measured based on three basic indices, i.e., network utilization, work-ahead, and the size of client buffer. Generally, network utilization is defined as the total bandwidth consumed by the streams transmitted over the communication network divided by the capacity of the network. The amount of time elapsed from the instant the media source starts to transmit data until the time the client can start to playback is defined as the work-ahead. The client buffer is used at the client to regulate burstiness. Sufficiently large buffer should be allocated to prevent it from being overflow or underflow. For live media, two general techniques are used to reshape the traffic of a media stream for better network utilization, i.e., to use work-ahead or to alter the characteristic of the encoder. In [9, 11], the authors present different prediction methods to estimate the size of a future frame based on the sizes of previous compress frames. As the network traffic can be predicted, VBR streams can be smoothed by work-ahead. In which, data can be sent ahead of the scheduled playback time to reduce the peak rate of a stream, thus more streams can be admitted to the network. If network congestion can be detected, Reibman and Berger [12] propose an approach to smooth network traffic by either slowing down the encoding speed or reducing the quality of encoding (e.g., by modifying the quantization matrix) and thus network traffic. Detailed discussions on the relations among the peak transmission rate, the amount of work-ahead, and the required buffer size can also be found in [10].

Stored media differs from live media in that the characteristics of media streams can be analyzed off-line to allow a better control for on-line transmission. In the following, some work on transporting stored media are recapped. In order to transmit a VBR media stream over a high-speed network via VBR service, a network transmission server must guarantees QoS either deterministically or statistically. A typical example of deterministic guaranteed server is the deterministic bounding interval dependent (*D-BIND*, for short)

traffic model proposed by Zhang and Knightly [17, 6]. The D-BIND approach models VBR traffic more accurately than those based on peak rate, average rate, and rate variance [17], and thus gives better network utilization. Based on the D-BIND approach, the renegotiated deterministic VBR scheme [6] (*RED-VBR*, for short) is introduced to gracefully renegotiate the varying D-BIND models during overload intervals. Because a renegotiation request could be rejected, renegotiation methods are considered as providing statistical QoS guarantee rather than deterministic. Recently, Salehi et al. [13] study the problem of using traffic smoothing to minimize rate variation of a VBR stream given a fixed work-ahead and client buffer. They show that an shortest-path based algorithm presented by Reibman and Berger [12] solves the minimum rate variability problem optimally. They also combine this minimum rate variability algorithm with a deterministic server, using D-BIND model in specific, and show that with optimal smoothing into a nominal buffer space, the number of admissible streams increases by as much as 100%.

In [4], Grosslauser and Keshav investigate the performance of CBR traffic in large-scale networks with many connections and switches to develop a framework for simulation. They conclude that network queuing delays is less than one cell times per switch even under heavy load. This is in contrast to VBR network services in which network queuing delay is usually one of the most significant factors in end to end delays. To transport VBR streams using CBR network services, M. Grassglauser, S. Keshav and D. Tse [5] present a renegotiation approach, called renegotiated CBR (*RCBR*, for short), to renegotiate varying CBR rates at different transmission intervals. Salehi et al. [13] combines a similar RCBR service with their minimum rate variability algorithm. However, these renegotiation based CBR approaches share the drawback of lacking deterministic guarantees. A deterministic guaranteed CBR service called constant-rate transmission and transport (CRTT) is presented by McManus and Ross in [10]. A CRTT server transmits video data to a client at constant rate without changes. The client starts to playback the first frame of video data after receiving a certain amount of data called *build up*. By choosing appropriate transmission rate and the size of *build up*, the authors show that the client buffer can be minimized by dynamic programming. The drawback of this transmission model which uses a constant rate for the entire connection time, as can be expected, is that the size of client buffer would be too large. As reported in [10], it requires 22.3MB memory to receive the movie *Star War* [15] although the buffer size is minimized. The authors also show that, by dividing the video stream into several segments and allowing transmission rate to vary among the segments, client buffer can be reduced. However, it still requires 2 MB memory buffer for *Star War*).

The CBR transmission model is known to have advantages in low overhead and easy management. In this paper, we study the problem of transporting VBR stored video (will also be referred to as video for brevity in

the rest of this paper) traffic using CBR network services. We are specifically interested in MPEG encoded store video traffic because of its popularity in both academic and industry. Our transmission scheme is unlike the CRTT server in which data is transmitted continuously at constant rate for the entire connection time of the traffic. There are two alternative transmission states in our transmission server, i.e., ON and OFF states [14]. Given a video stream $\mathcal{V} = \{f_1 f_2 \dots f_n; T_f\}$ where f_i is the ith frame, the server sends data at rate r during ON state and sends no data during OFF state. A transmission schedule γ refers to an $(2m+3)$-tuple $\gamma = \{r; t_0, t_1, \dots, t_{2m+1}\}$, i.e., it starts sending data by rate r at time t_0, pauses at t_1, resumes sending data at t_2, etc. Playback at constant frame rate $1/T_f$ (frames per second) of the video stream from the client buffer becomes a constraint in designing the transmission schedule γ, where T_f denotes the time interval between adjacent video frames. The goal of our design is to compute a transmission schedule γ for a video stream such that its utilization $u = T_{ON}/T_c$ is maximized and the required client buffer size is minimized, where T_{ON} and T_c denote the total ON time and the total connection time respectively.

Our solution to this problem is based on the following algorithms. Given a video stream $\mathcal{V}$ and a target transmission rate r, the algorithm *Lazy* computes the minimum client buffer size $\hat{b}_{\mathcal{V}}(r)$ and the minimum work-ahead $\hat{w}_{\mathcal{V}}(r)$. Using these information, algorithm *Aggressive* then computes a transmission schedule γ such that its network utilization $\hat{u}_{\mathcal{V}}(r)$ is maximized. A summary of the applied notations is shown in Table 1. Algorithms *Lazy* and *Aggressive* are shown to have $O(n)$ time complexity, where n denotes the number of frames in the video stream $\mathcal{V}$. Insertion of the OFF states in our transmission model prevents the server from sending excessive data and thus decreases the required buffer size. Intuitively, it would take a longer period to transmit the video stream. However, as the peak rate transmissions are utilized in the ON states as long as possible, experimental results on several benchmarks of MPEG streams show that the bandwidth utilization of our algorithm is comparable with or better than the conventional schemes. Furthermore, our scheme requires smaller buffer size. Note that, as our transmission rate is fixed with only ON/OFF two states, the network transmission can be guaranteed to be unit delay for queuing. Our applied network management and resource sharing processes are more easier than the conventional approaches with varying CBR rates in different transmission intervals.

As different clients may have different architecture environments, it is critical to give clients the flexibility to determine buffer size. Besides, the other parameters like work-ahead and transmission rate that should be also decided by both client and server. Note that the functions $\hat{b}_{\mathcal{V}}(r)$, $\hat{w}_{\mathcal{V}}(r)$ and $\hat{u}_{\mathcal{V}}(r)$ for different peak transmission rate r are monotonically non-increasing functions, which can be computed in brute force by algorithms with time complexity $O(n^3)$. In [7], we have proposed an $O(n \log n)$ algorithm to compute

$\mathcal{V} = \{ f_1 f_2 \ldots f_n; T_f \}$	a video stream
f_k	the kth video frame, and also used to denote its size
n	number of frames in V
T_f	time interval between two adjacent frames
$CPF(t)$	cumulative playback function of stream $\mathcal{V}$
$F_k = \sum_{i=1}^{n} f_i$	cumulative frame size
$\|\mathcal{V}\| = \sum_{k=1}^{n} f_k$	total size of the video stream V
r	peak transmission rate
$\gamma = \{r; t_0, \ldots, t_m\}$	transmission schedule
m	total number of ON and OFF states
Γ	cumulative transmission function
$\gamma_s = \{x_1, \ldots, x_{n'}\}$	transport stream
n'	total number of transport frames
T_{ON}	total ON time during transmission
T_c	total connection time
$u_{\mathcal{V},\gamma} = T_{ON}/T_c$	utilization
w_γ	work-ahead of transmission schedule γ
$b_{\mathcal{V},\gamma}$	client buffer size for playback of stream $\mathcal{V}$ using transmission schedule γ
$\hat{w}_{\mathcal{V}}(r)$	minimum work-ahead for playback of stream $\mathcal{V}$ using a transmission schedule with rate r
$\hat{b}_{\mathcal{V}}(r)$	minimum client buffer size for playback of stream $\mathcal{V}$ using a transmission schedule with rate r
$\hat{u}_{\mathcal{V}}(r)$	maximum utilization for playback of stream $\mathcal{V}$ using a transmission schedule with rate r
Lazy	algorithm for computing minimum buffer and work-ahead
Aggressive	algorithm for computing maximum utilization subject to minimum buffer and work-ahead constraints
MB	minimum buffer problem
MW	minimum work-ahead problem
$BBMU$	bounded buffer maximum utilization problem
$MBMU$	minimum buffer maximum utilization problem

Table 1: Notations used in this paper.

these functions. In this paper, we consider only the optimization of VBR stored video transmission on CBR channel for a given peak transmission rate. The proposed schemes can be easily extended to minimize the rate variability for VBR video transmission. They can be also applied to resolve the buffer-constrained transmission problem with the minimum peak transmission rate for a given buffer size. The remainder of this paper is organized as follows. In Section 2, problem definitions and related mathematical formulations are given. We present the algorithms and optimality proofs in Section 3. In Section 4, empirical studies are presented. Concluding remarks and future directions are given in section 5.

2 Problem Definition

In this paper, we study the problem of transmitting VBR video via CBR network services to support video on demand and other similar applications. Our primary design goal is to minimize client buffer size, to maximize network utilization, and to minimize the work-ahead of the transmission schedule. Let's start with considering end-to-end transmission of stored video streams from a video server to a client across the network as shown in figure 1. For each stream, data are first retrieved from the storage subsystem and stored in the

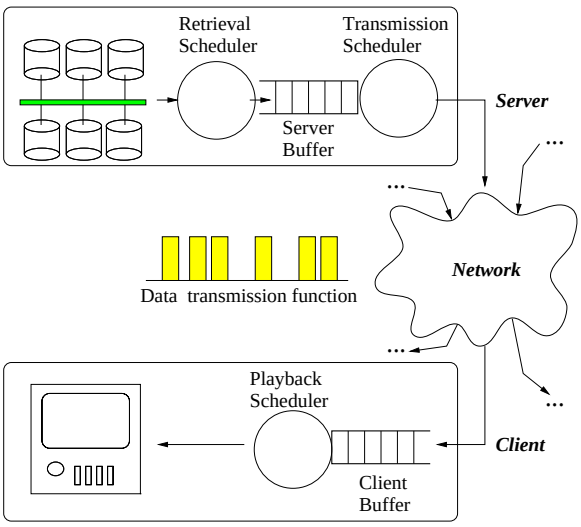

Figure 1: The considered end-to-end transmission, in which, data streams are delivered from server to client across the network.

server buffer as dictated by a retrieval scheduler, such as GSS or SCAN-EDF [1]. This scheduler guarantees that data are retrieved before the time they are scheduled to transmit. The transmission scheduler get data from the server buffer and sends them to the client at proper timing. The design of transmission scheduler is the focus of this paper, and its goal is to minimize system cost including the cost of the receiving client in terms of memory usage and the cost of network transmission in terms of network utilization and management complexity. At the client side, incoming data stream are temporarily stored in the bounded-capacity client buffer where server work-ahead is enabled. Video data stored at the client buffer is then retrieved and playback frame by frame periodically by the playback scheduler. Note that, if a frame arrives late or is incomplete in the client buffer at its scheduled playback time, unpleasant jittery effects are then perceived by the audience. To avoid jittery playback, the transmission schedule must always be ahead of the playback schedule subject to the condition that it does not overflow the client buffer. The amount of time between the start of transmission and the start of playback is denoted as *work-ahead*.

To formalize the above discussions, first, we'll assume that the lower network layers corrects transmission errors automatically. Thus, transmission errors do not occur at the transportation layer which is the focus of this paper. We also assume that end to end network delay from the server to a client is zero. Though the results presented in this paper are based on this zero-delay assumption, it is not difficult to show that these results also apply to the cases in which network delay is upper bounded by a certain constant. Interested

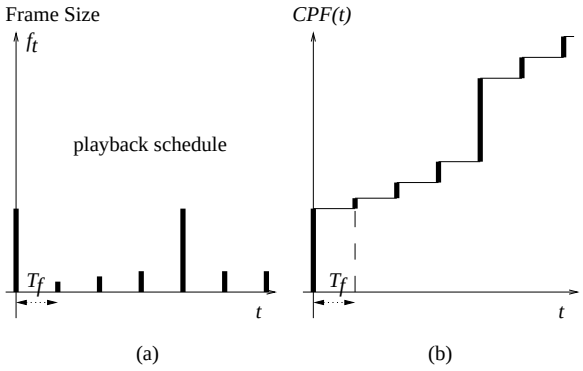

Figure 2: (a) A VBR video stream and (b) its cumulative playback function.

readers may refer to [4] for further discussions. Let's start with the definition of *cumulative playback function* and cumulative transmission function.

Definition 1 *Given a video stream* $\mathcal{V} = \{f_0 f_1 \dots f_{n-1}; T_f\}$ *and its cumulative frame size* $F_i = F_{i-1} + f_i$, *where* $0 \leq i < n$ *and* $F_{-1} = 0$, *the cumulative playback function CPF*(t) *is defined as*

$$CPF(t) = \begin{cases} 0, & t < 0 \\ F_i, & iT_f < t < (i+1)T_f \text{ where } 0 \leq i < n, \\ F_n, & t > (n-1)T_f. \end{cases}$$

Note that, in this paper, we assume that a video stream is always playback at $t = 0$. As defined previously in section 1, a video stream $\mathcal{V}$ consists of a sequence of n video frames $\mathcal{V} = \{f_0 f_1 \dots f_{n-1}; T_f\}$ with each frame being separated by a constant time interval T_f, where f_i denotes the ith frame. An example of a video stream $\mathcal{V}$ and its cumulative playback function $CPF(t)$ is given in figure 2.

The conventional CBR network transmission is known to posses the advantages in network management. Unfortunately, for stored video applications with inherent VBR traffic, a much larger buffer is required at the receiving client to avoid data overrun as well as data under-run. In this paper, based on the proposed LA method, a type of network transmitter which transmits data at peak rate r on and off is studied. That is, the network transmitter either sends data at its peak rate r or sends nothing at all. This transmission scheme, usually referred to as *on-off* transmission in literatures (see, e.g., [14]), unlike CBR transmission, does not flood the client buffer all the time. Specifically, it does not have to send excess data when the client has sufficient supply for its future use, but, at the cost of a lower level of network utilization. Fortunately, our study shows that, by appropriately designing the transmission algorithm with respect to a specific video stream, the client buffer size is decreased dramatically at a nominal level of network utilization even when comparing with the best known VBR traffic smoothing and admission control algorithms.

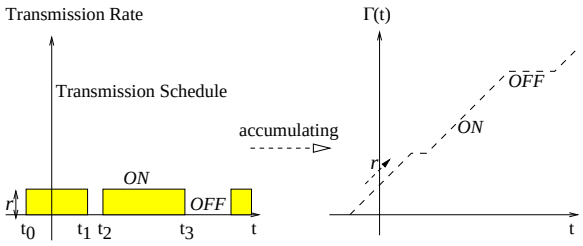

Figure 3: A simple example to illustrate the transmission behavior of the ON-OFF CBR model.

Definition 2 *A transmission schedule is defined as* $\gamma = \{r; t_0, t_1, \ldots, t_{2m+1}\}$. *A transmitter following* γ *transmits data at rate* r *in each time slot* $(t_{2i}, t_{2i+1}), 0 \leq i \leq m$, *and sends nothing in each time slot* $(t_{2i-1}, t_{2i}), 0 < i \leq m$. *A transmission schedule can thus also be written as*

$$\gamma(t) = \begin{cases} r, & t \in (t_{2i}, t_{2i+1}), 0 \leq i \leq m; \\ 0, & otherwise. \end{cases}$$

The cumulative transmission function $\Gamma(t)$ *is defined as the integration of* $\gamma(t)$, *i.e.,* $\Gamma(t) = \int_{-\infty}^{t} \gamma(x)dx$. *We also denote* $\Gamma(kT_f)$ *as* Γ_k.

According to this definition, the cumulative transmission function $\Gamma(t)$ is defined as the amount of data sent by the transmitter up to time t, and the transmission schedule $\gamma(t)$ as the derivative of the cumulative transmission function with respect to t. A typical example of the transmission schedule and its cumulative transmission function $\Gamma(t)$ is shown in figure 3. Note that $\Gamma(t)$ also denotes the amount of data received by the receiver up to time t because zero network delay is assumed. Note that, though zero delay assumption is made, results presented in this paper can be easily generalized to networks with bounded delay as shown in [17]. Important properties of Γ are monotonicity and continuity as stated below. We omit the proof of this lemma and leave it to interested readers.

Lemma 1 *The cumulative transmission function* $\Gamma(t)$ *is continuous and monotonically non-decreasing.*

In order for a video stream to be transmitted from a video server and start jitter-free playback at time $t = 0$, the transmission schedule must be always ahead of the playback of the video stream by a sufficient amount of work-ahead w. This notion can be expressed in terms of the cumulative transmission function and the cumulative playback function of the video stream as follows.

Definition 3 *A transmission schedule $\gamma = \{r; t_0, \ldots, t_{2m+1}\}$ is said to be feasible for transporting a video stream $\mathcal{V}$ if its cumulative transmission function $\Gamma(t) \geq CPF(t)$ and $\Gamma((n-1)T_f) = |\mathcal{V}|$, where $CPF(t)$ is the cumulative playback function of $\mathcal{V}$. The value $w = -t_0$ is referred to as the work-ahead of γ.*

Note that when we say that a transmission schedule is feasible for transporting a video stream $\mathcal{V}$ in this paper, what it really means is that under this transmission schedule, the video transmission server will transmit data on time for jitter-free playback of video stream $\mathcal{V}$ as the client starting at time $t = 0$. According to the definition of $CPF(t)$ and monotonicity of $\Gamma(t)$, the following lemmas are self-evident. Their proofs are left for interested readers.

Lemma 2 *A transmission schedule $\gamma = \{r; t_0, \ldots, t_{2m+1}\}$ is said to be feasible for video stream $\mathcal{V}$ if $\Gamma_k \geq F_k, 0 \leq k \leq n-2$, and $\Gamma_{n-1} = F_{n-1}$.*

Consider a transmission schedule γ which is feasible for transporting a video stream $\mathcal{V}$, since $\Gamma(t) - CPF(t)$ is the transmitted data temporarily stored in the client buffer, buffer size requirement at the receiving client can be computed as follows.

Definition 4 *Given a transmission schedule $\gamma = \{r; t_0, \ldots, t_{2m+1}\}$, the client buffer size $b_{\mathcal{V},\gamma}$ for a receiving client to playback video stream $\mathcal{V}$ is $b_{\mathcal{V},\gamma} = \max\{\Gamma(t) - CPF(t); \forall t\}$.*

Obviously, buffer size $b_{\mathcal{V},\gamma}$ is no smaller than the maximum frame size $\max\{f_i\}$, and is no larger than the size $\mathcal{V}$ of the video stream.

Lemma 3 *The buffer size $b_{\mathcal{V},\gamma}$ required by a client to playback a video stream $\mathcal{V} = \{f_0, f_1, \ldots, f_{n-1}; T_f\}$ by receiving it from a video server according to a feasible transmission schedule γ is $b_{\mathcal{V},\gamma} = \max\{\Gamma_k - F_{k-1} | 0 \leq k \leq n\}$, where $\Gamma_k = \Gamma(kT_f)$ and F_{k-1} denotes the cumulative frame size of $\mathcal{V}$ at time $t = (k-1)T_f$.*

That is, instead of computing over the entire continuous time domain for the required buffer size, it suffices to compute the required buffer size for $m+1$ sample points. An example to illustrate the cumulative playback function of a video stream, a feasible cumulative playback function, the work-ahead and the buffer size at the client is depicted in figure 4. If the client has a limited amount of buffer space b, then obviously $b \geq b_{\mathcal{V},\gamma}$ must hold. Alternatively, it can also be written as follows.

Definition 5 *A transmission schedule $\gamma = \{r; t_0, \ldots, t_{2m+1}\}$ is said to be feasible for transporting a video stream $\mathcal{V}$ with bounded buffer size b at the receiving client if $F_k \leq \Gamma_k \leq \min\{|\mathcal{V}|, F_{k-1} + b\}$, where $\Gamma_k = \Gamma(kT_f)$ and $|\mathcal{V}|$ denotes the total size of the video stream $\mathcal{V}$.*

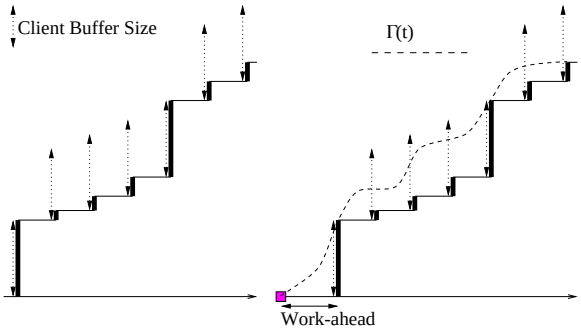

Figure 4: A simple example illustrating the relations among the cumulative transmission function, the cumulative playback function, the work-ahead and the client buffer size.

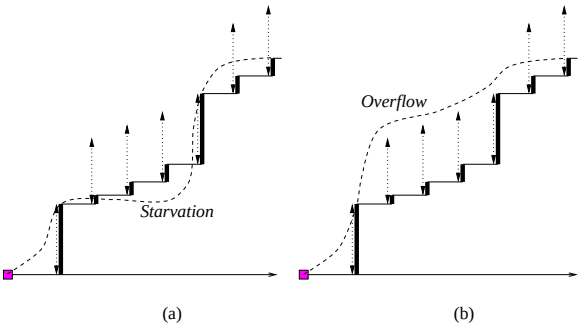

Figure 5: (a) The starvation condition and (b) the overflow condition for the cumulative transmission function with a bounded client buffer.

Figure 5 shows the starvation condition and the overflow condition for the cumulative transmission function with a bounded client buffer at the receiving client.

The network utilization $u_{\mathcal{V},\gamma}$ is defined as the ratio of the video stream size $|\mathcal{V}|$ and the amount of data which could be transmitted using the constant bandwidth allocated to the video stream for the entire network connection time.

Definition 6 *The network utilization $u_{\mathcal{V},\gamma}$ of transmitting video stream $\mathcal{V} = \{f_0, f_1, \ldots, f_{n-1}; T_f\}$ using transmission schedule $\gamma = \{r; t_0, t_1, \ldots, t_{2m+1}\}$ is defined as the ratio of the total amount of data transmitted over the maximum amount of data that can be transmitted at rate r during the time period (t_0, t_{2m+1}), i.e., $u_{\mathcal{V},\gamma} = |\mathcal{V}|/(r * T_c) = T_{ON}/T_c$ where $T_{ON} = \sum_{i=0}^{m}(t_{2i+1} - t_{2i})$ and $T_c = t_{2m+1} - t_0$ denote the total ON time and the total connection time respectively.*

We are now ready to introduce the problems studied in this paper. First, given a constant rate r and a video stream $\mathcal{V}$, we study the problem of designing a transmission schedule such that buffer size and work-ahead are minimized. Then, among the class of transmission schedules minimizing buffer size and work-ahead, we study the problem of designing a transmission schedule such that utilization of its allocated bandwidth r is maximization.

Definition 7 *Given a constant r and a video stream $\mathcal{V}$, the following problems are defined:*

MB Problem *in a* minimum buffer *problem (MB problem for short), the objective is to design a transmission schedule γ of rate r to transport $\mathcal{V}$ such that the client buffer size $b_{\mathcal{V},\gamma}$ is minimized; denote the minimum buffer size as $\hat{b}_{\mathcal{V}}(r)$.*

MW Problem *: In a* minimum work-ahead *problem (MW problem for short), the objective is to design a transmission schedule γ of rate r to transport $\mathcal{V}$ such that the work-ahead of γ is minimized; denote the minimum work-ahead as $\hat{w}_{\mathcal{V}}(r)$.*

BBMU Problem *: In a* bounded buffer, maximum utilization problem *(BBMU problem for short), the objective is to design a transmission schedule γ of rate r to transport $\mathcal{V}$ such that the client buffer size $b_{\mathcal{V},\gamma}$ is no greater than b and the network utilization $u_{\mathcal{V},\gamma}$ is maximized; denote the maximum utilization as $\hat{u}_{\mathcal{V},\lfloor}(r)$.*

MBMU Problem *: In a* bounded buffer, maximum utilization problem *(MBMU problem for short), the objective is to design a transmission schedule γ of rate r to transport $\mathcal{V}$ such that the client buffer size $b_{\mathcal{V},\gamma}$ is no greater than $\hat{b}_{\mathcal{V}}(r)$ and the network utilization $u_{\mathcal{V},\gamma}$ is maximized, where $\hat{b}_{\mathcal{V}}(r)$ denotes the minimum buffer size as obtained from solving the MB problem; denote the maximum utilization as $\hat{u}_{\mathcal{V}}(r)$.*

Note that, we are also interested in the computation of the characteristic curves $\hat{b}_{\mathcal{V}}(r)$, $\hat{w}_{\mathcal{V}}(r)$, and $\hat{u}_{\mathcal{V}}(r)$, for all values of r. It is shown in our another paper [7].

3 Algorithm LA

In this section, we present *algorithm L* to solve the problems MB and MW, and *algorithm LA* to solve the problems BBMU and MBMU. *Algorithms L* and *LA* are shown to solve the corresponding problems optimally. *Algorithm LA* has two phases. In the first phase, it runs *algorithm L* to compute $\hat{b}_{\mathcal{V}}(r)$ and $\hat{w}_{\mathcal{V}}(r)$ for a given rate r. It then runs *algorithm A* to compute an optimum transmission schedule $\hat{\gamma}$ such that it achieves maximum utilization $\hat{u}_{\mathcal{V}}(r)$ using minimum client buffer size $\hat{b}_{\mathcal{V}}(r)$. *Algorithms L* and *LA* both runs in $O(n)$ time. We also show that the characteristic curves $\hat{b}_{\mathcal{V}}(r), \hat{w}_{\mathcal{V}}(r)$, and $\hat{u}_{\mathcal{V}}(r)$ can be computed in $O(n^3)$ time by computing them on $O(n^2)$ critical values of r. Detailed description of these computations are presented below.

3.1 Minimize Buffer and Work-ahead by Algorithm L

Given video stream $\mathcal{V}$ with cumulative frame size $F_k, 0 \leq k < n$, let's start with the definition of a sequence $L_k, k \geq 0$. We'll show later that L_k defines the cumulative transmission function of a feasible transmission schedule to transport $\mathcal{V} = \{f_0 f_1 \ldots f_n; T_f\}$ such that client buffer size and transmission work-ahead are both minimized.

Definition 8 *Let $F_k, k \geq 0$ denote cumulative frame size of video stream $\mathcal{V}$. The sequence $L_k, k \geq 0$ is defined as follows:*

$$\begin{cases} L_i = |\mathcal{V}| & \forall i \geq n-1, \text{and} \\ L_k = \max\{F_k, L_{k+1} - rT_f\} & \forall 0 \leq k < n-1. \end{cases}$$

The following lemma is a direct consequence of definition 8.

Lemma 4 *Let $F_k, k \geq 0$, denote cumulative frame size of video stream $\mathcal{V}$ and $L_k, k \geq 0$ denote the sequence defined by definition 8. If $L_{k+1} - L_k < rT_f$, then we have $L_k = F_k$.*

We define the transmission schedule $\gamma_L = \{r; t_0, t_1, \ldots, t_{2m+1}\}$ as follows. Given a constant r and a video stream $\mathcal{V}$, the lazy transmission schedule $\gamma_L = \{r; t_0, t_1, \ldots, t_{2m+1}\}$ with on-off CBR transmission is computed by *Algorithm L* presented in figure 6. A example illustrating the computation of the lazy transmission schedule γ_L is shown in Fig. 7. Before showing that $\hat{w}_{\mathcal{V}}(r) = -t_0$ and $\hat{b}_{\mathcal{V}}(r) = b_{\mathcal{V},\gamma_L}$. Let's consider the following lemma.

Lemma 5 *The function $\gamma_L(t)$ defined by the transmission schedule γ_L as computed by Algorithm L satisfies $\gamma_L(kT_f) = L_k$, and is feasible for transporting $\mathcal{V}$. Moreover, the parameters $t_k, 0 \leq k \leq 2m+1$, of $\gamma_A(t)$ are all distinct.*

Algorithm L
input: a constant r and a video stream $\mathcal{V}$
output: a transmission schedule γ_L

Compute $L_i = |\mathcal{V}| \quad \forall i \geq n-1$;
Compute $L_k = \max\{F_k, L_{k+1} - rT_f\} \quad \forall 0 \leq k < n-1$;
$t_0 = -L_0/r$; $j = 1$;
for $k = 0$ to $n-1$ do
    begin
        if $L_{k+1} - L_k < rT_f$ then
            begin
                $t_j = L_k$;
                $t_{j+1} = L_{k+1} - (L_{k+1} - L_k)/r$;
                $j = j + 2$;
            end
    end

Figure 6: Algorithm L.

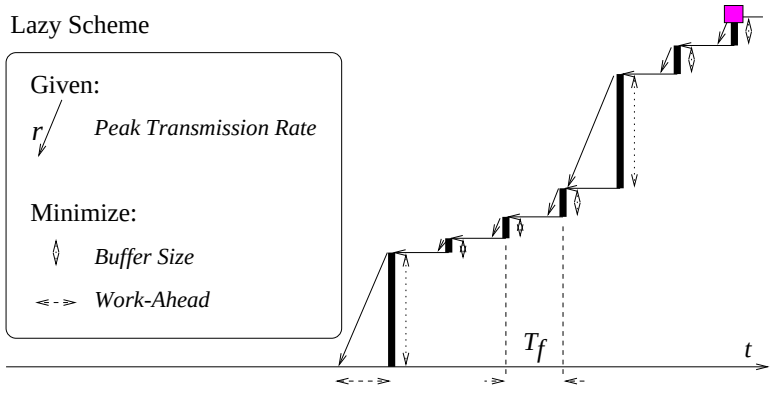

Figure 7: A simple processing example of the lazy scheme.

We left the proof to interested readers.

The transmission schedule γ_L captures the intuition of transmitting data as late as possible. So, we could expect that the client buffer size $b_{\mathcal{V},\gamma_L}$ is minimum. The formal proof is presented as follows.

Theorem 1 *$b_{\mathcal{V}}(r) = \max_{k=0}^{n}\{L_k - F_{k-1}\}$ is the minimum client buffer size for transporting video stream $\mathcal{V}$ at peak transmission rate r.*

Proof:

First, by lemma 3, we have $b_{\mathcal{V},\gamma} = \max_{k=0}^{n}\{L_k - F_{k-1}\}$. Denote $i, 0 \leq i \leq n$, as a point at which $L_i - F_{i-1} = b_{\mathcal{V},\gamma_L}$. Let's assume that transmission schedule γ' defines a smaller client buffer size than does γ_L in transporting stream $\mathcal{V}$. Then we have $\Gamma'_i < L_i$.

Let's consider the following two cases:

(1) $L_{i+1} - L_i < rT_f$. By lemma 4, we have $L_i = F_i$. Since $\Gamma'_i < L_i$, it contradicts the feasibility condition for γ'.

(2) Otherwise, we have $L_{i+1} - L_i = rT_f$. Let i' denotes the smallest integer such that $i' > i$ and $L_{i'+1} - L_i < rT_f$. That is, $L_{i'} = F_{i'}$. But, then we have $\Gamma'_{i'} \leq \Gamma'_i + (i' - i)T_f < L_i + (i' - i)T_f = L_{i'} = F_{i'}$. It again contradicts the feasibility condition for γ'.

We thus conclude that the client buffer size of γ_L is minimum. Q.E.D.

Though it is shown that the transmission schedule γ_L uses minimum client buffer in transporting a stream $\mathcal{V}$ in theorem 1. We can also use a similar proof to assert that γ_L is a *minimum transmission schedule* for transporting $\mathcal{V}$ in the sense that for every feasible transmission schedule $\gamma_{\mathcal{V}}$ with rate r and cumulative transmission function $\Gamma(t)$, we have $\Gamma(kT_f) \geq L_k, \forall 0 \leq k < n$.

Corollary 1 *Let $\Gamma(t)$ denote the cumulative transmission function of a feasible transmission function $\gamma_{\mathcal{V}}$ of rate r of a video stream $\mathcal{V}$. We have $\Gamma(kT_f) \geq L_k, \forall 0 \leq k < n$.*

Thus, at time $t = 0$, L_0 is the minimum amount of data to pre-fill client buffer so that jitter-free playback can be guaranteed. At a give peak transmission rate r, the time $w(r) = L_0/r$ is then the minimum work-ahead for jitter-free playback.

Corollary 2 *To transport video stream $\mathcal{V}$ at a given rate r, work-ahead of transmission schedule γ_L is a minimum, i.e., $\hat{w}_{\mathcal{V}}(r) = L_0/r$.*

3.2 Optimizing Network Utilization by Algorithm LA

Though the transmission schedule γ_L is shown to minimize client buffer size and transmission work-ahead without violating peak rate constraints. On the other hand, corollary 2 also show that γ_L is a minimum transmission schedule. Specifically, the end of its transmission is synchronized with the playback of the last frame. Thus, the end of γ_L is the latest among all the feasible transmission schedules for transporting $\mathcal{V}$. Furthermore, as data are transmitted as late as possible, it is less robust against network jitter. In this subsection, by using the minimum client buffer size $\hat{b}_{\mathcal{V}}(r)$ and the minimum work-ahead $\hat{w}_{\mathcal{V}}(r)$ as constraints, we study the problem of designing a transmission schedule for $\mathcal{V}$ such that it finishes its transmission as early as possible. Let's start with the definition of the sequence $A_k, k \geq 0$.

Definition 9 *Let $F_k, k \geq 0$, denote cumulative frame size of video stream $\mathcal{V}$. The sequence $A_k, 0 \leq k < n$ is defined as follows:*

$$\begin{cases} A_0 = L_0, & \forall i \geq n-1, \textit{ and} \\ A_k = \min\{|\mathcal{V}|, F_k + \hat{b}_{\mathcal{V}}(r), A_{k-1} + rT_f\}, & \forall k > 0. \end{cases}$$

The following lemma is a consequence of definition 9.

Lemma 6 *Let $F_k, k \geq 0$, denote cumulative frame size of video stream $\mathcal{V}$ and $A_k, k \geq 0$, denote the sequence defined by definition 9. We have*
(1) if $A_k - A_{k-1} < rT_f$ and $A_k \neq |\mathcal{V}|$, then $A_k = F_k + \hat{b}_{\mathcal{V}}(r)$;
(2) $A_k \geq F_k$.

Similar to the definition of γ_L, we define the transmission schedule γ_A as follows. Given a constant r and the sequence $A_k, 0 \leq k < n$, the aggressive transmission schedule $\gamma_A = \{r; t_0, t_1, \ldots, t_{2m+1}\}$ is computed by *Algorithm A* presented in figure 8. A simple example to present the processing of the aggressive scheme is shown in Fig. 9.

Note that, *Algorithm A* can be used to compute a transmission schedule with client buffer size $b' > \hat{b}_{\mathcal{V}}(r)$ and peak transmission rate r. As γ_A keeps on transmitting data to the client as long as the client buffer is not full, it performs better control against network transmission jitters. The network utilization is maximized and the changes of on-off states are minimized. Similar to lemma 5, we also have the following lemma for γ_A.

Algorithm A
input: a constant r, client buffer size $\hat{b}_{\mathcal{V}}(r)$ and work-ahead $\hat{w}_{\mathcal{V}}(r)$ constraints, and a video stream $\mathcal{V}$
output: a transmission schedule γ_A

Compute $A_0 = L_0, \ \forall i \geq n-1$;
Compute $A_k = \min\{|\mathcal{V}|, F_k + \hat{b}_{\mathcal{V}}(r), A_{k-1} + rT_f\}, \ \forall k > 0$;
$t_0 = -A_0/r; \ \ j = 1$;
for $k = 0$ to $n-1$ do
    begin
        if $A_{k+1} - A_k < rT_f$ then
            begin
                $t_j = A_k + (A_{k+1} - A_k)/r$;
                $t_{j+1} = A_{k+1}$;
                $j = j + 2$;
            end
    end

Figure 8: Algorithm A.

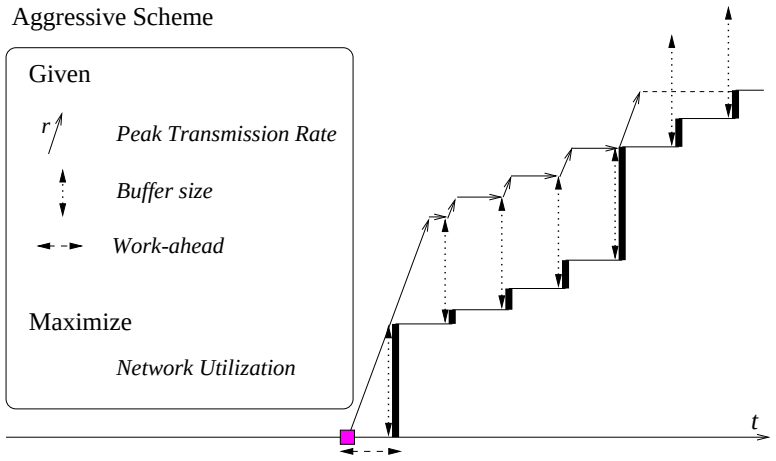

Figure 9: An example of γ_A.

Algorithm LA

Step 1: Input a video $\mathcal{V} = f_0 f_1 \ldots f_{n-1}$ and a peak transmission rate r

Step 2: Compute the cumulative frame size $F_i = F_{i-1} + f_i$, $0 \leq i < n$, and $F_{-1} = 0$.

Step 3: Apply *Algorithm L* to compute γ_L with $L_i = \max\{F_i, L_{i+1} - rT_f\}, 0 \leq i \leq n-2$, and $L_{n-1} = |\mathcal{V}|$. The buffer size and the work-ahead can be computed by $\hat{b}_{\mathcal{V}}(r) = \max_{i=0}^{n-1}\{L_i - F_i\}$ and $\hat{w}_{\mathcal{V}}(r) = L_0/r$, respectively.

Step 4: Apply *Algorithm A* to compute γ_A with $A_i = \min\{|\mathcal{V}|, F_i + \hat{b}_{\mathcal{V}}(r), A_{i-1} + rT_f\}, 1 \leq i \leq n-1$ and $A_0 = L_0$.

Figure 10: Algorithm LA.

Lemma 7 *The function $\gamma_A(t)$ defined by the transmission schedule γ_A as computed by Algorithm A satisfies $\gamma_A(kT_f) = A_k$, and is feasible for transporting $\mathcal{V}$. Moreover, the parameters $t_k, 0 \leq k \leq 2m+1$, of $\gamma_A(t)$ are all distinct.*

Algorithm LA as presented in figure 10 is a combination of *Algorithm L* and *Algorithm A*. It is obvious that *Algorithm LA* runs in $O(n)$ time. We are now ready to present the following theorem.

Theorem 2 Algorithm LA *solves MBMU problem in $O(n)$ time.*

Proof: To show this, it suffices for us to show that the connection time of transmission schedule γ_A is a minimum. We assume that contrary. Since we have shown that γ_A has the earliest time to start transmission, thus, there is a transmission schedule γ' such that its completion time iT_f is earlier than that of γ_A, i.e. $\Gamma'_i = |\mathcal{V}|$. Let's consider the following three cases:

(1) There is some $i' \leq i$ such that $A_{i'} - Fi' = \hat{b}_{\mathcal{V}}(r)$. Without loss of generality, we assume that i' is the largest integer satisfying these conditions. Then we have $\Gamma'_{i'} - F_{i'} \geq \Gamma'_i - (i-i')T_f - F_{i'} > A_i - (i-i')T_f - F_{i'} = A_{i'} - F_{i'} = \hat{b}_{\mathcal{V}}(r)$. Which contradicts the client buffer size constraint.

(2) For each $i' \leq i$, $A_{i'} - Fi' < \hat{b}_{\mathcal{V}}(r)$. Furthermore, there is some $j', i < j' < n'$ such that $A_{j'} - Fj' = \hat{b}_{\mathcal{V}}(r)$, where $n'T_f$ denotes the completion time of γ_A. Then we have $\Gamma'_i - F_i > A_{j'} - F_i \leq A_{j'} - F_{j'} = \hat{b}_{\mathcal{V}}(r)$. Again, it contradicts the client buffer size constraint.

(3) $A_{n'} - Fn' = \hat{b}_{\mathcal{V}}(r)$, while for each $i' < n', A_{i'} - Fi' < \hat{b}_{\mathcal{V}}(r)$. In this case, we can show that the connection time of γ' is no less than that of γ_A. The details are left for interested readers. Q.E.D.

Note that, the network utilization may be further improved by increasing client buffer size at the same transmission rate r. One can observe this by fixing r and work-ahead $\hat{w}_{\mathcal{V}}(r)$, as a result, the completion time of the transmission schedule is moved forward. We note that the above arguments also generalize to arbitrary buffer size constraints with $b \geq \hat{b}_{\mathcal{V}}(r)$. That is, we have the following corollary.

Corollary 3 *The BBMU Problem can be solved in $O(n)$ time.*

Note that, the proposed method can be easily extended to minimize the rate variability for VBR video transmission if an optimal smoothing aggressive transmission algorithm is applied. Whenever a buffer-constrained lazy transmission algorithm is applied, the proposed method can be used to minimize the peak transmission rate for a given buffer size.

3.3 Computing the Characteristic Curves

Given a video stream $\mathcal{V}$ and a peak transmission rate r, we present *algorithm LA* (as shown in figure 10) to construct a transmission schedule with minimum buffer size, minimum work-ahead, and maximum network utilization. The time complexity is $O(n)$ where n is the number of frames in the video stream. Using different peak rate to transmit a video stream requires different required buffer size, work-ahead, and utilization. To facilitate resource management and admission control for QoS guarantees, we need to explore the relations among the client buffer size, work-ahead and network utilization, and how they vary as a function of transmission rate r. Let's first consider the boundary cases with small and large values of r. When r is close to 0, the client buffer size $\hat{b}_{\mathcal{V}}(r)$ is roughly the same as $|\mathcal{V}|$, i.e., to pre-store the entire video stream. In this case, network bandwidth is fully utilized, i.e., $u = 100\%$. The rate of change in client buffer size depends on where the bottleneck of buffer size requirement lies. That is, if i denotes the largest integer, $0 \leq i \leq n-1$, satisfying $L_i - F_{i-1} = \hat{b}_{\mathcal{V}}(r), 0 \leq i \leq n-1$, then the rate of change in client buffer size with respect to r is proportional to $n-1-i$. Network utilization falls below 100% when transmission rate r is smaller than $r_1 = \min_{j=0}^{n-1}\{(F_{n-1} - F_j)/(n-j-1)T_f\}$. At this moment, we also notice that the transmission schedule is decomposed into two ON periods, i.e., $(0, iT_f)$ and $((i+1)T_f, (n-1)T_f)$, and one OFF period. Between $r = 0$ and $r = r_1$, $\hat{b}_{\mathcal{V}}(r)$, as a function of r, changes its slope if and only if there is some value $r_1^1, 0 < r_1^1 < r_1$, such that the bottleneck moves backward to say a position $t = i'T_f$, $i < i' < n-1$, and $r_1^1 = (F_{i'-1} - F_{i-1})/(i'-i)T_f$. The rate r_1^1 is computed by considering the fact that $L_i - F_{i-1} = L_{i'} - F_{i'-1}$ and $L_{i'} = L_i + (i'-i)r_1^1 T_f$.

On the other extreme, if rT_f is larger than the maximum frame size, i.e., $r \geq r_R = \max_{i=0}^{n-1} f_i/T_f$, then the client buffer size is exactly the maximum frame size $f_M \max_{i=0}^{n-1} f_i$. As r decreases below r_R , the client buffer size non-decrease above f_M. Client buffer size remains the same as f_M even when transmission rate r is larger than r_R. In this case, the transmission schedule γ_L is decomposed into n ON periods and $n-1$ OFF periods. According to the above discussions, a rough sketch of the minimum buffer size $\hat{b}_{\mathcal{V}}(r)$ as a function of transmission rate r is depicted in figure 11(a). Give the minimum client buffer size, figure 11(b) sketches

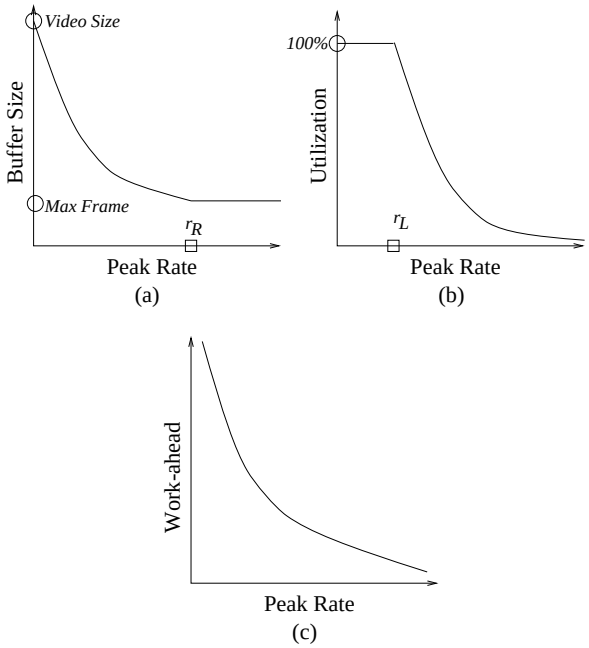

Figure 11: (a) The relation between the minimum client buffer size and transmission rate r. (b) The relation between network utilization and transmission rate r subject to minimum client buffer size constraint. (c) The relation between the work-ahead and transmission rate r.

the relation between the maximum network utilization and transmission rate r. A sketch of the minimum work-ahead as a function of transmission rate r is given in figure 11(c).

As briefly mentioned above, to compute the characteristic curves $\hat{b}_{\mathcal{V}}(r)$, $\hat{w}_{\mathcal{V}}(r)$, and $\hat{u}_{\mathcal{V}}(r)$ of a video stream $\mathcal{V}$, it suffices to compute them over a finite set of critical values of $r^c = (F_i - F_j)/(i-j), 0 \leq i, j \leq n-1$. These $O(n^2)$ critical values either contribute as the critical rate to decompose a continuous ON period into two smaller ones or to swap the bottleneck of buffer size requirement from one point to the other. This naive algorithm takes $O(n^3)$ time to compute these characteristic curves. In [7], an $O(n \log n)$-time algorithm is presented.

4 Experimental Results

In this section, *Algorithm LA* presented in this paper is examined under guaranteed service on several VBR-encoded MPEG traces. Some of them are captured by a number of researchers [3, 8, 6, 12] who are kind enough to share them with us. The others are down-loaded from the anonymous ftp archive [16]. In Table 2, we list the examined VBR traces and their related statistics (the number of frames, the maximum frame size and the average frame size). As the time complexity of *Algorithm LA* is O(n), its implementation is simple

and efficient. By considering an over 2 hours long video with 174136 frames, it takes only 5 seconds (on a Sun Workstation) to compute the transmission schedule.

Stream Name	No. of Frames	Max Frame Size (KB)	Avg Size (KB/f)
Star War	174136	22.62	1.90
Princess Bride	167766	29.73	4.89
CNN News	164748	30.11	4.89
Wizard of OZ	41760	41.89	5.09
Advertisements	16316	10.08	1.86
Lecture	16316	6.14	1.37

Table 2: Statistics of the Examined VBR Traces.

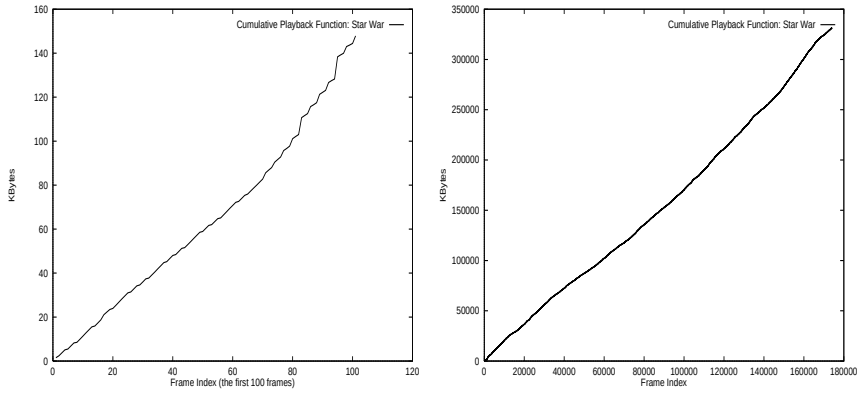

Figure 12: The cumulative playback function (CPF) of the Star War movie trace.

The first trace examined in our analysis is a *Star War* MPEG1 movie. Its cumulative playback function (CPF) is shown as figure 12. The average frame size is 1.9 KB with the maximum frame size 22.62 KB and the minimum frame size 0.28 KB. As the frame size variability is large, it requires 23 MB memory buffer and 37sec work-ahead for the transmission schedule obtained by the CRTT approach [10]. Our experiments show that, by applying *Algorithm LA*, the *Star War* video stream can be jitter-free transmitted using 6 MB memory buffer and 26 ms work-ahead. The obtained network utilization is nearly 80% with 0.47 Mbps transmission rate. Figure 13 shows the minimum buffer size and the maximum network utilization obtained by the proposed *LA* approach for different peak transmission rates. As the required memory buffer and work-ahead are minimized by *Algorithm LA*, they are smaller than those obtained in the CRTT approach as shown in figure 14 (left).

More details on the relation between the required buffer size and the obtained network utilization is shown in figure 14 (right). Note that, client buffer size increases as the network utilization increases. In

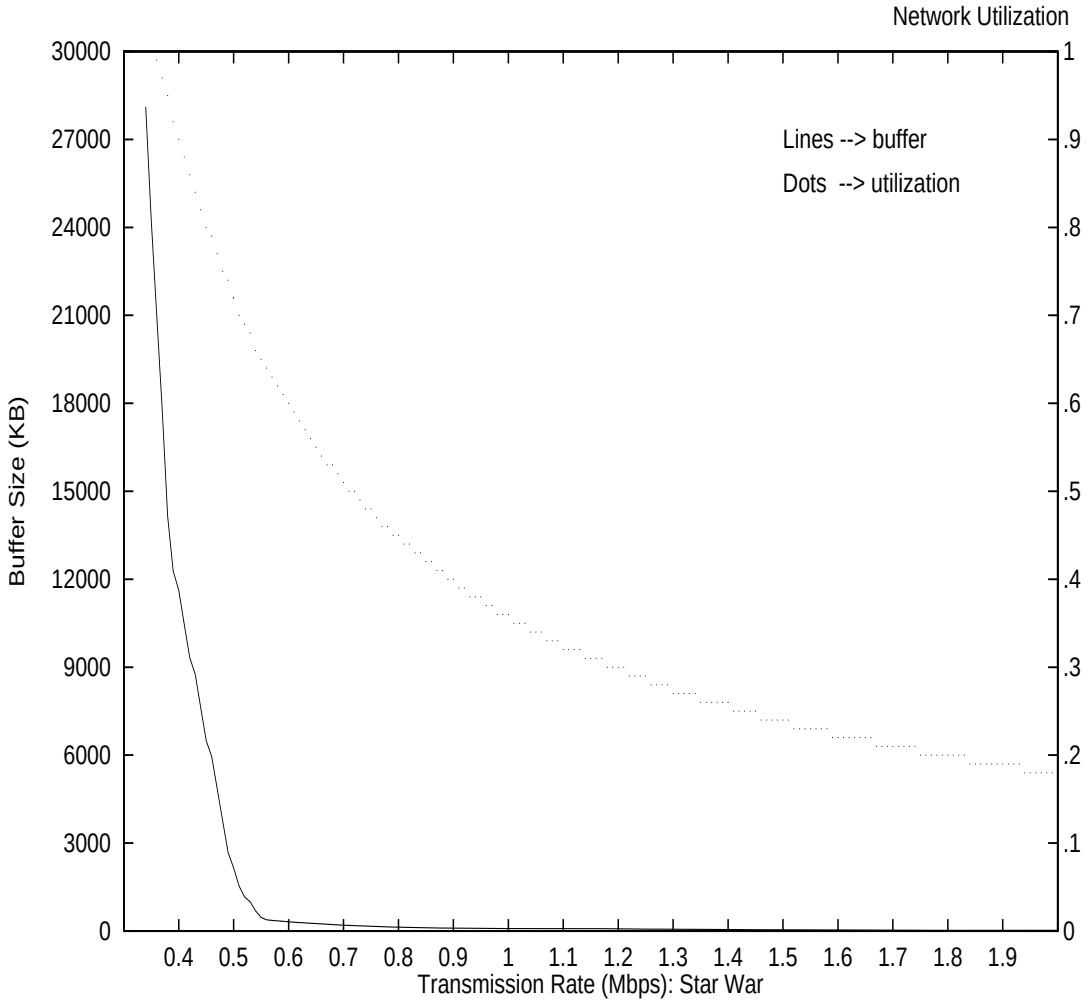

Figure 13: Buffer and network utilization obtained by LA for different transmission rates.

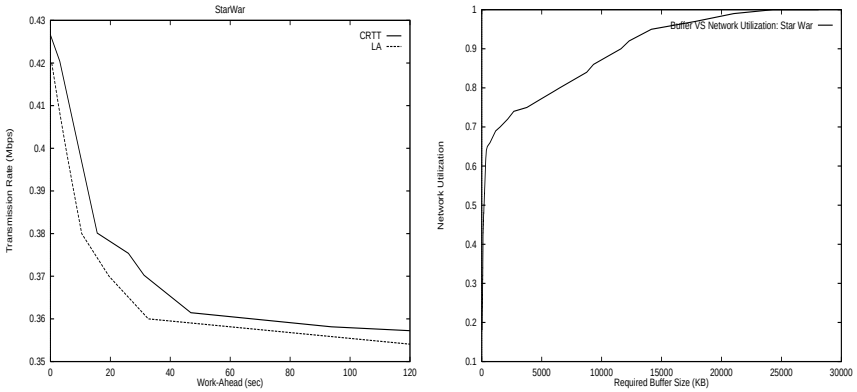

Figure 14: Star War: (left) The comparisons of the proposed approach and the CRTT approach by the required work-ahead and transmission rate. (right) The relation between the required buffer size and the obtained network utilization.

Star War, with additional 5 MB memory buffer, network utilization increases from 80% to 90% and it supports more requests. Thus, when a client request to view a movie (such as *Star War*), it should send the available buffer size to the server. The server looks up the relation between the required buffer size and the possible network utilization to allocate transmission bandwidth. Depending on the client's preference, the transmission schedule can specify memory-conscious or communication-conscious. If the transmission schedule is memory-conscious, the server will determine the memory buffer as small as possible while the network utilization is still under the client constraints. If the transmission schedule is transmission-conscious, the server will try to maximize the network utilization with the constraints of the available buffer size.

Based on the relation between client buffer size and network utilization, the session set-up protocol can be as simple as a request-reply. For instance, a viewer wants to watch the *Star War* movie and the memory buffer he can afford is 2 MB with 1.5 Mbps transmission rate. If the viewer is memory-conscious, the server may determine a transmission schedule of 43 KB memory and 1.5 Mbps transmission rate. On the other hand, if the viewer is communication-conscious, then the transmission schedule could be of 2 MB memory and 0.5 Mbps transmission rate. If the viewer can only afford 50 KB memory buffer and 1.0 Mbps for the peak transmission rate, the server will reject the request. It can be shown that there is no guaranteed service for this request. The admission policy is simple in this design that the server checks the request against its available resources, and decide to admit or not.

When evaluated by different long video streams such as the *Princess Bride* movie and the *CNN News*, *Algorithm LA* also obtains good results. Figure 15 shows the cumulative playback function of the 90 minutes long *Princess Bride* movie with 167766 frames. The minimum buffer size and the maximum network

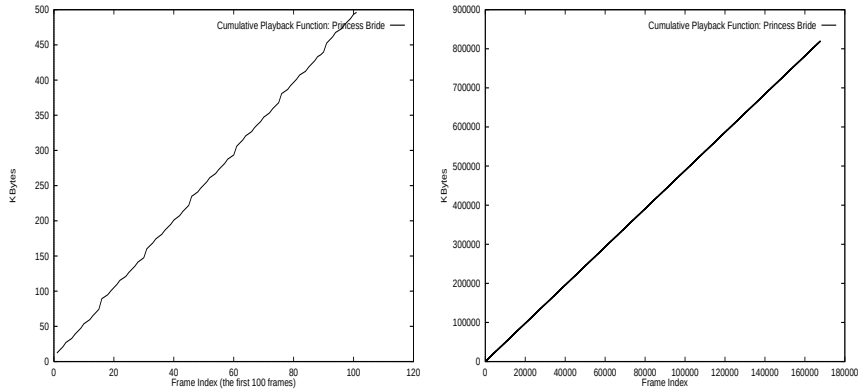

Figure 15: The cumulative playback function (CPF) of the Princess Bride movie trace.

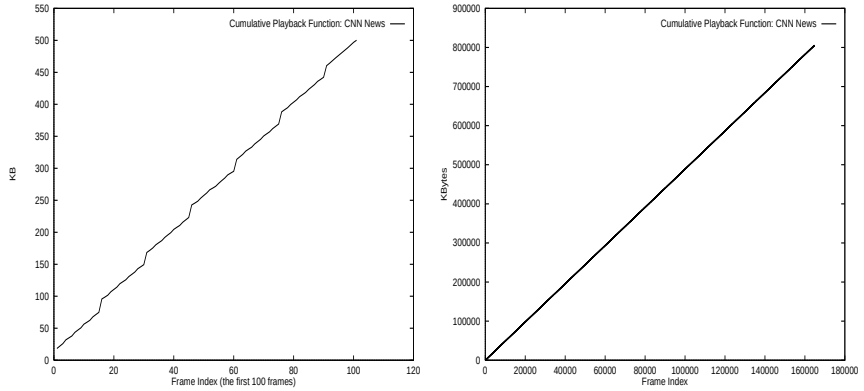

Figure 16: The cumulative playback function (CPF) of CNN News video trace.

utilization obtained by the proposed LA approach for different peak transmission rates is shown in figure 20. The cumulative playback function of *CNN News* video trace is shown in figure 16 at 30 fps frame rate. Figure 21 shows the minimum buffer size and the maximum network utilization obtained by the proposed LA approach for different peak transmission rates. *Algorithm LA* guarantees the services for these two video traces by only 38 KB memory buffer at 1.16 Mbps transmission rate. As shown in figure 18, the obtained network utilization is nearly 100%. On an OC-3 link providing 127.155 Mbps for video data transport, the transmission schedule can support over 109 video streams. By considering the 23 minutes long *Wizard of Oz* video trace as shown in figure17, the transmission schedule for all these 41760 frames can be guaranteed using 6 MB memory buffer and 1.5 Mbps transmission rate. The network utilization is over 80% and over 84 video streams can be supported on an OC-3 link. Figure 22 shows the minimum buffer size and the maximum network utilization obtained by the proposed LA approach for different peak transmission rates.

The *Advertisements* and the *Lecture* video traces are all with 16316 frames in length. Both of them are 10 minutes long at 30 fps frame rate with a 160x120 frame size. By applying *Algorithm LA*, the *Advertisements*

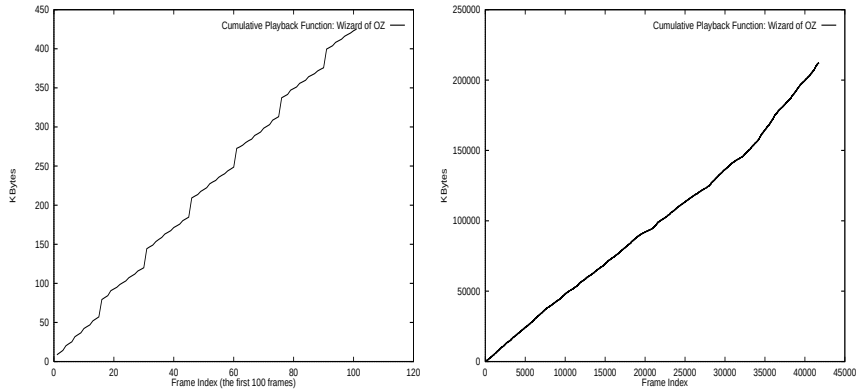

Figure 17: The cumulative playback function (CPF) of the Wizard of OZ movie.

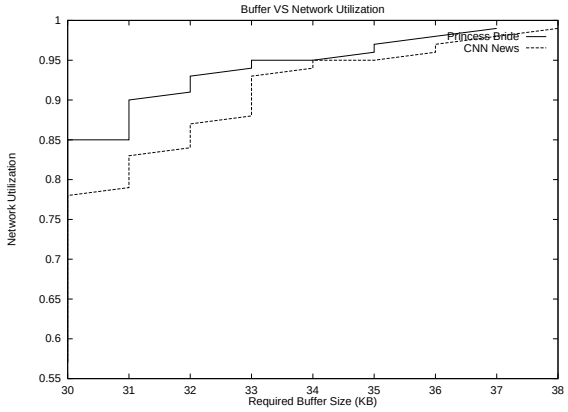

Figure 18: The relation between the required buffer size and the obtained network utilization of CNN News and Princess Bride.

video stream can be jitter-free transmitted using 420 KB memory buffer with 513 ms work-ahead. The obtained network utilization is 75% with 0.59 Mbps transmission rate. Figure 23 shows the minimum buffer size and the maximum network utilization obtained by the proposed LA approach for different peak transmission rates. The *Lecture* video trace, a lecture showing the speaker and his slides along with zooming and panning, can be jitter-free transmitted under 570 KB memory buffer with 77 ms work-ahead. The obtained network utilization is 89% with 0.36 Mbps transmission rate. Figure 24 shows the minimum buffer size and the maximum network utilization obtained by the proposed LA approach for different peak transmission rates. The relation between the required buffer size and the obtained network utilization is shown in figure 19. *Algorithm LA* has proved to maximize the network utilization and minimize the required memory buffer for a given peak transmission rate. Assume that the transmission work-ahead is 100 ms, *Algorithm LA* can achieve nearly 70% network utilization for the transmission schedule of *Advertisements*.

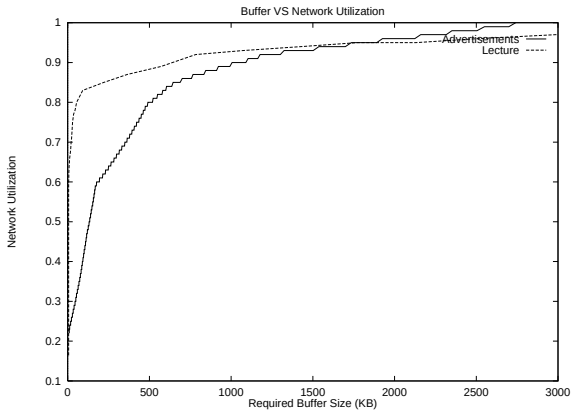

Figure 19: The relation between the required buffer size and the obtained network utilization of Advertisements and Lecture.

It is higher than the D-BIND [6] approach which achieves only 26% network utilization. Although Salehi et. al.'s approach [13] can also achieve the similar network utilization, it requires more than 1 MB (1159 KB) of memory buffer. While *Algorithm LA* requires only 316 KB memory buffer. More comparisons for the relation between the work-ahead and the network utilization is shown in figure 25. More test results to the video streams *MTV*, *Indiana*, *Racing*, *EricClapton*, *UnderSiege*, *Space*, *Bird* and *LAsmog* are shown in [2].

5 Conclusion

In this paper, we present a CBR transmission algorithm for jitter-free VBR stream playback. As the kernel of our transmission algorithm, *Algorithm LA* is show to minimize client buffer size and work-ahead with the maximum network utilization. We have explored *Algorithm LA* for transmitting several stored video from a server to clients across the high-speed network. By considering a presented transmission rate, *Algorithm LA* has achieved better buffer utilization than that of the CRTT approach. Without complicating the management of network and server resources, our approach requires much smaller memory buffer than does CRTT transport. When comparing with the optimal smoothing method [13], *Algorithm LA* smoothes VBR streams for the minimum buffer size and the maximum network utilization. Note that, depending on the preference of the system, the optimality could be in terms of the minimum buffer size or the maximum network utilization. This approach shows great flexibility to allow various client machines to set up their best transmission contexts. For example, the optimality may be in network utilization that drive the network utilization as large as possible with the cost of extra buffer. The fixed rate CBR counts on pre-processing of video frames in order to decide the needed resources for jitter-free playback. We can assume a certain

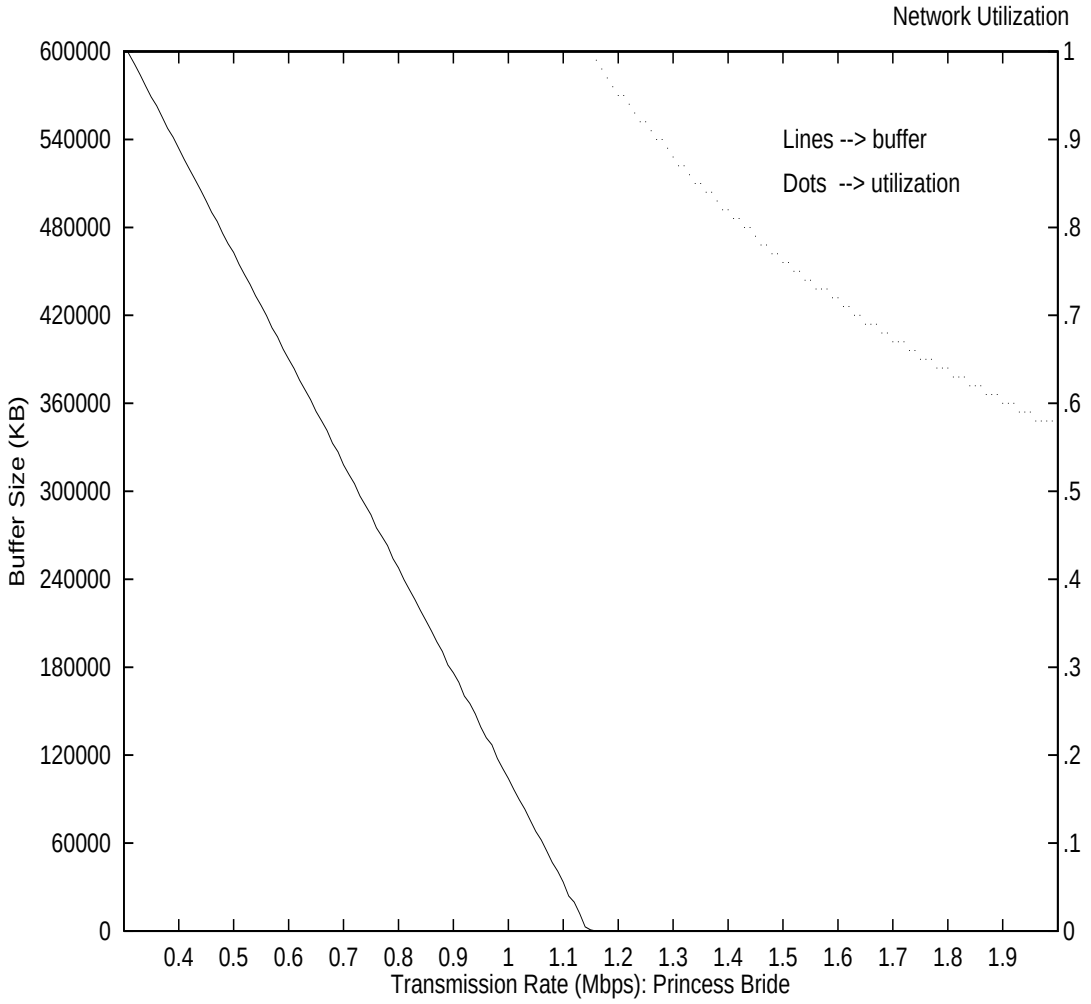

Figure 20: Buffer and network utilization obtained by LA for different transmission rates.

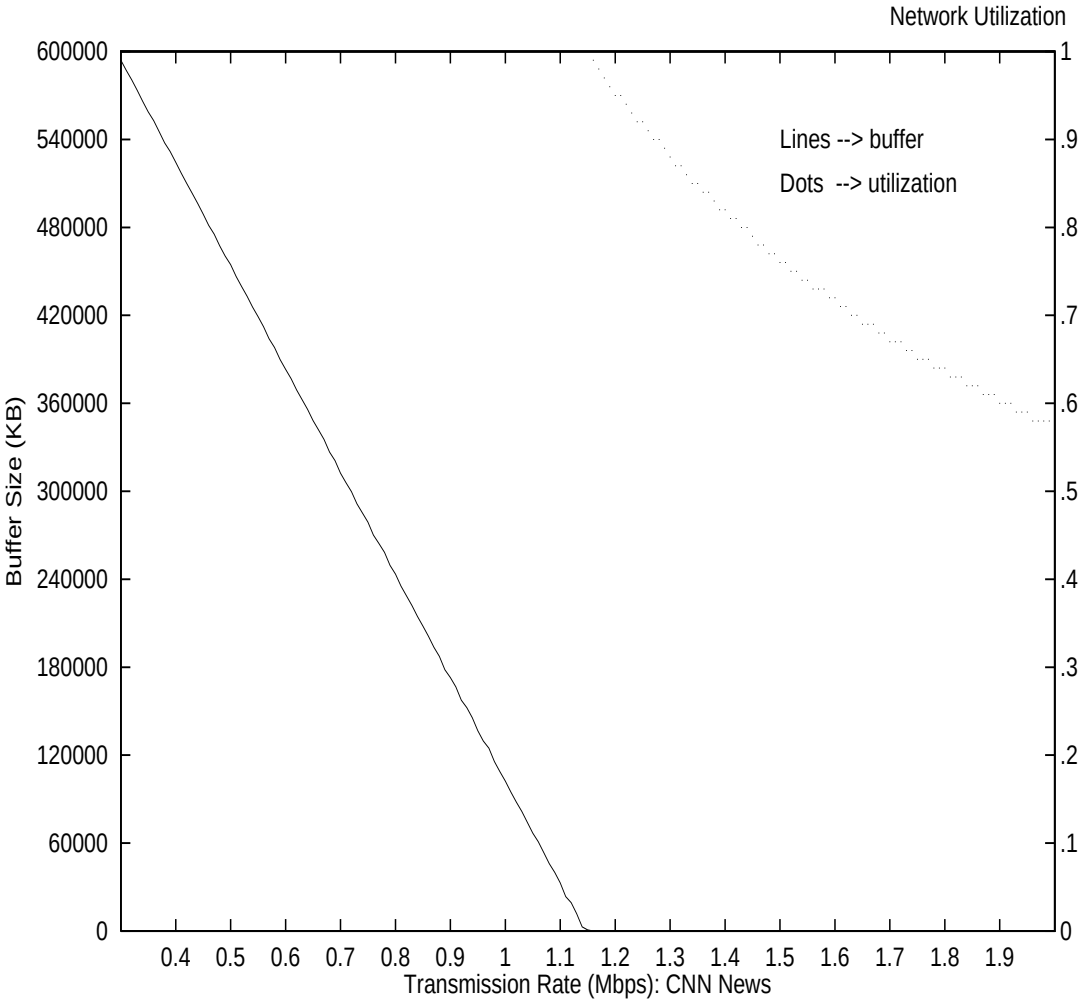

Figure 21: Buffer and network utilization obtained by LA for different transmission rates.

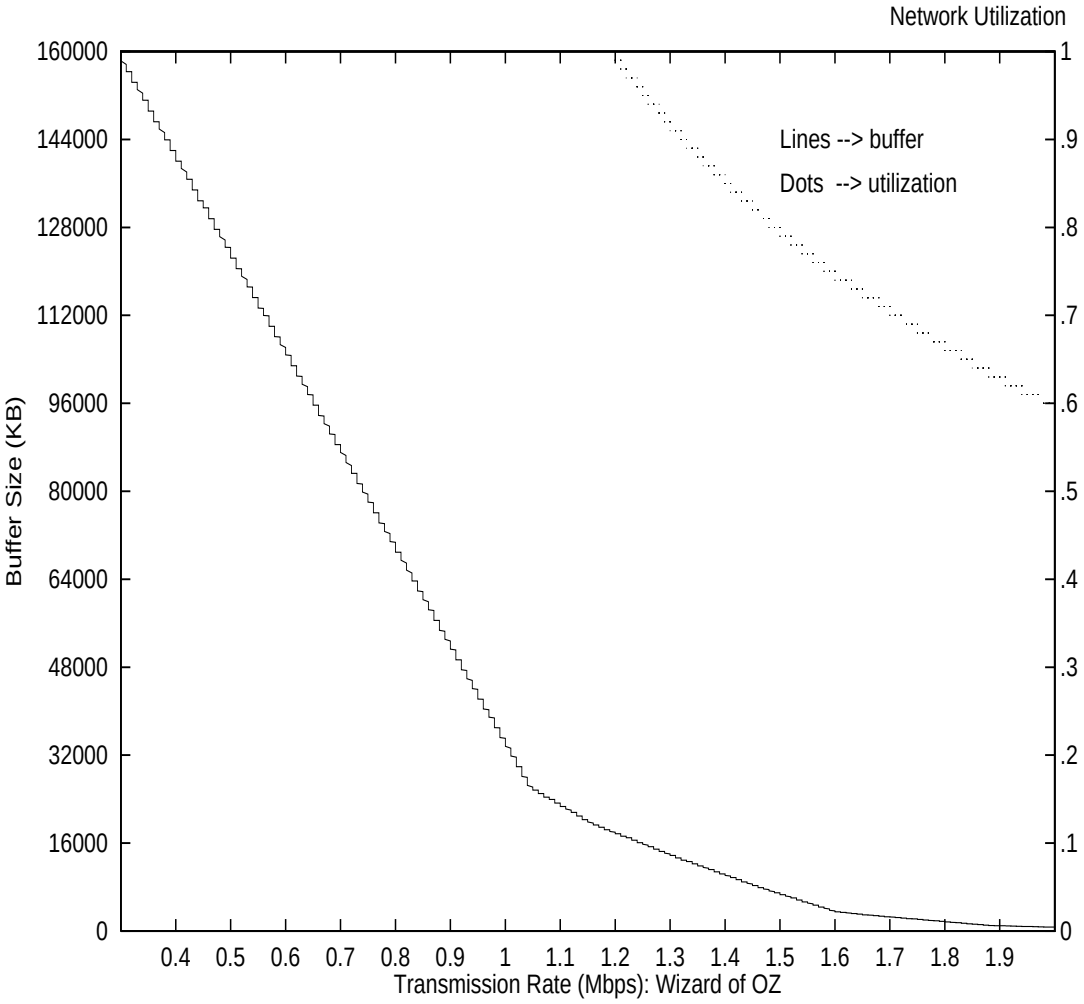

Figure 22: Buffer and network utilization obtained by *LA* for different transmission rates.

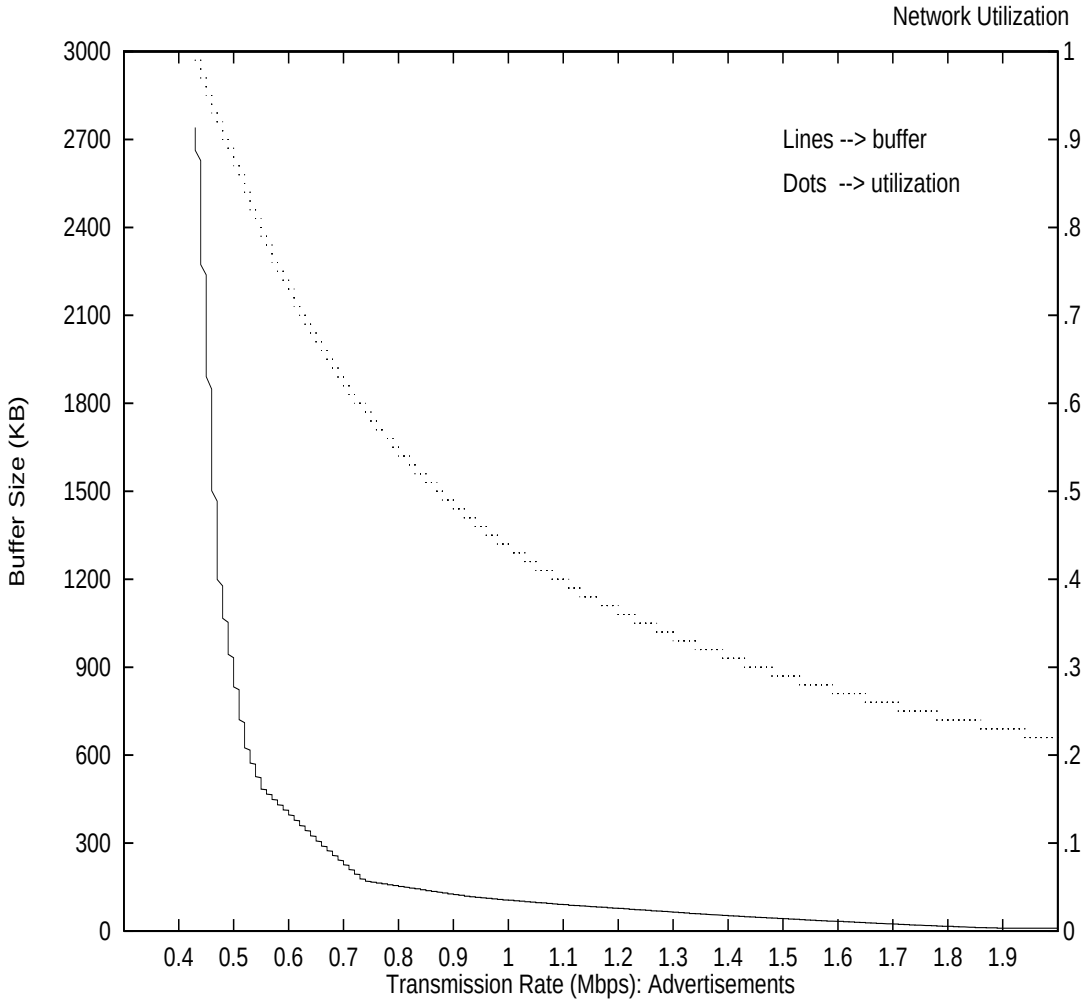

Figure 23: Buffer and network utilization obtained by *LA* for different transmission rates.

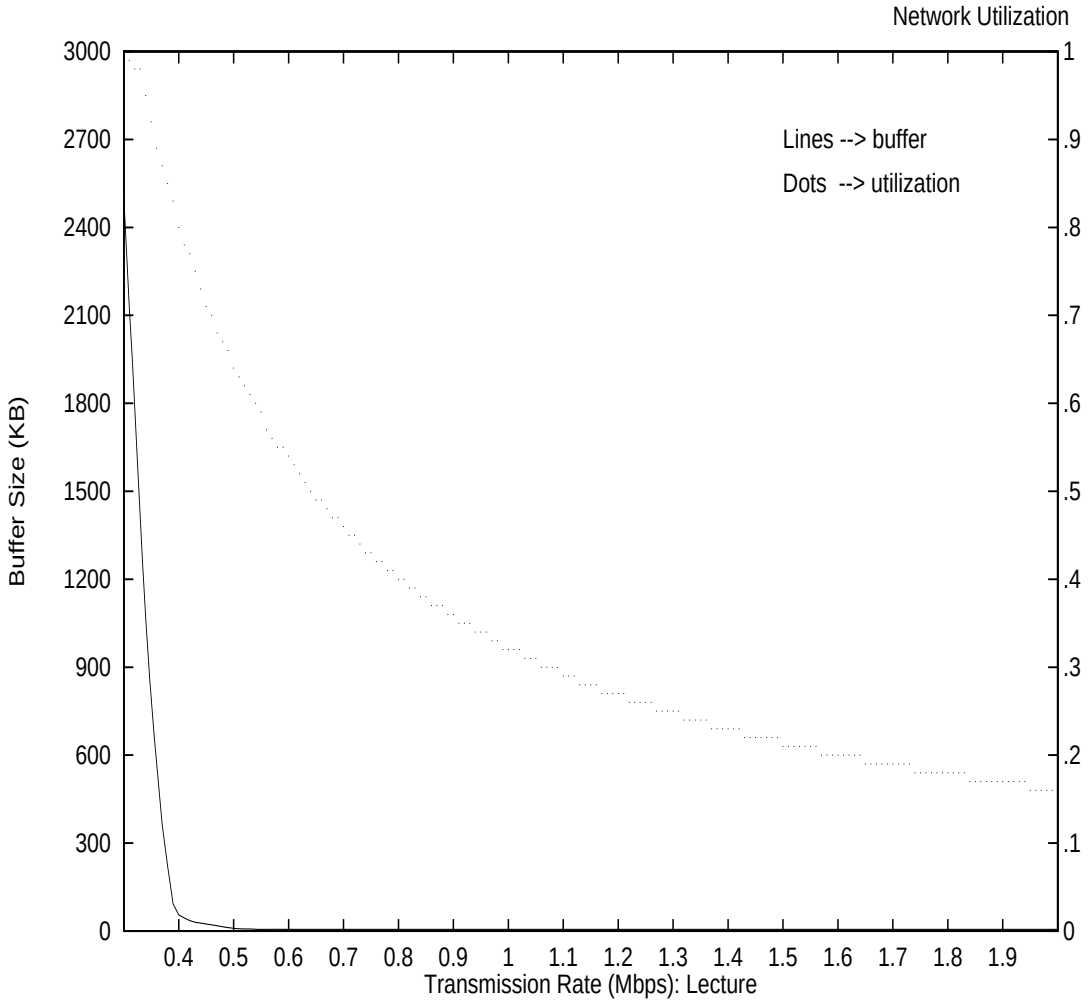

Figure 24: Buffer and network utilization obtained by LA for different transmission rates.

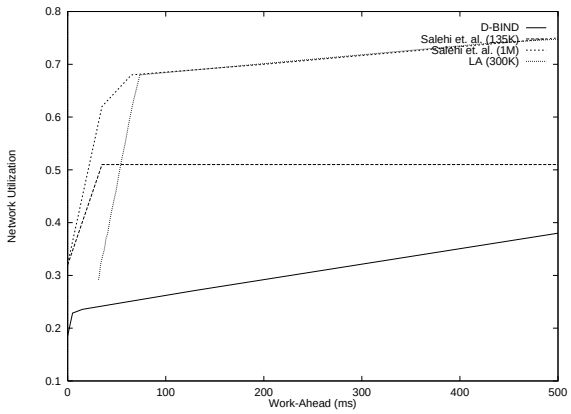

Figure 25: The relation between the work-ahead and the obtained network utilization of Advertisements.

transmission rate and study the issue of how the server can use the network bandwidth most effectively toward reducing the required buffer size. Our approach is practical, efficient, and flexible in supporting jitter-free video playback. In our future work, we will develop algorithms to calculate optimal parameters for supporting VCR features.

References

[1] E. Chang and A. Zakhor. Scalable video data placement on parallel disk arrays. In *IS&T/SPIE Symposium on Electronic Imaging Science and Technology*, February 1994.

[2] Ray-I Chang, Meng Chang Chen, Ming-Tat Ko, and Jan-Ming Ho. Fixed rate CBR transmission for jitter-free VBR video playback. 1996.

[3] M. Garrett and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM*, pages 269–280, August 1994.

[4] M. Grossglauser and S. Keshav. On CBR service. In *Proc. IEEE INFOCOM*, March 1996.

[5] M. Grossglauser, S. Keshav, and D. Tse. RCBR: a simple and efficient service for multiple time-scale traffic. In *Proc. ACM SIGCOMM*, August 1995.

[6] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang. Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic. In *Proc. ACM SIGMETRICS*, pages 47–55, may 1995. D-BIND model.

[7] Ming-Tat Ko, Jan-Ming Ho, Meng Chang Chen, and Ray-I Chang. An O($n$log$n$) algorithm to compute rate-buffer curve for the admission control of VBR video transmission on CBR channel. 1996.

[8] M. Krunz and H. Hughes. A traffic model for mpeg-coded VBR streams. In *ACM SIGMETRICS*, pages 47–55, May 1995.

[9] S. S. Lam, S. Chow, and D. K. Y. Yau. An algorithm for lossless smoothing of MPEG video. In *Proc. ACM SIGCOMM*, 1994.

[10] J. M. McManus and K. W. Ross. Video on demand over ATM: Constant-rate transmission and transport. In *Proc. IEEE INFOCOM*, March 1996.

[11] T. Ott, T. V. Lakshman, and A. Tabatabai. A scheme for smoothing delay-sensitive traffic offered to ATM networks. In *Proc. IEEE INFOCOM*, 1992.

[12] A. R. Reibman and A. W. Berger. Traffic descriptors for VBR video teleconferencing over ATM networks. *IEEE/ACM Transactions on Networking*, June 1995.

[13] J. D. Salehi, Z. L. Zhang, J. F. Kurose, and D. Towsley. Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing. In *Proc. ACM SIGMETRICS*, 1996.

[14] K. Sohraby. On the theory of general ON-OFF source with applications in high-speed networks. In *IEEE INFOCOM*, pages 401–410, 1993.

[15] MPEG traces. data available via anonymous ftp. FTP ftp::/thumper.bellcore.com.

[16] MPEG videos. data available via anonymous archive. URL http::/www.eeb.ele.tue.nl/mpeg/.

[17] H. Zhang and E. W. Knightly. A new approach to support delay-sensitive VBR video in packet-switched networks. In *Proc. 5th Workshop on Network and Operating Systems Support for Digital Audio and Video*, 1995.