Novel optimal real-time admission control and buffer management for the Video-On-Demand (VOD) service

Cheng-Shong Wu[a], Shi-Wei Lee[a], Wei-Kuo Lin[a] and Frank Yeong-Sung Lin[b]

[a] *Department of Electrical Engineering, National Chung Cheng University, Chia-Yi, 621, Taiwan*
E-mail: ieecsw@ccunix.ccu.edu.tw;p8310@eeipx2.ee.ccu.edu.tw
[b] *Department of Information Management, National Taiwan University, Taipei, Taiwan*
E-mail: yslin@im.ntu.edu.tw

In order to meet the Quality of Service (QoS) requirements of the VOD service, and, on the other hand, to maximize the system throughput (revenue), it is essential that the video call request admission control algorithm be carefully designed. In this paper, in addition to the Single Segment (SS) admission control first described in [7], we propose two new types of admission control schemes called Segmental Re-tuned (SR) admission control and Multiple Segment (MS) admission control for variable-bit-rate video streams under various video server architectures. The basic approach to the algorithm development is first to formulate each problem as a mathematical problem and then to identify special structures and properties for such formulations so that optimal real-time algorithms can be developed. In computational experiments, it is shown that the proposed algorithms for the considered video server architectures, compared with the traditional admission control scheme based upon the peak frame size, typically achieve over 175–200% improvement in the system throughput. In addition, the new proposed MS scheme performs 5–25% better than SS and SR schemes.

1. Introduction

The Video-On-Demand (VOD) service [11] has become feasible due to the availability of a number of enabling technologies, such as the MPEG (Moving Picture Experts Group) coding scheme, the ADSL (Asymmetric Digital Subscribers Line) technique and the ATM (Asynchronous Transfer Mode) technology. The VOD service can provide viewers with the quality and control similar to those of the VCR playback of rental programs but with a lot more flexibility and convenience. As such, many information providers, network providers, and Customer Premises Equipment (CPE) vendors are actively realizing the service.

For continuous retrieval of video data, it is essential that video information be available at the display device by the time of its playback. We refer to this as the *continuity requirement*. In order to satisfy the *continuity requirement* for each active video stream, admission control schemes should be carefully designed. Three types of admission control schemes have been proposed. They are strict admission control schemes [1,8,13], predictive admission control schemes [2,4], and pre-analyzed admission control schemes [7].

The objective of strict admission control schemes is to guarantee that the continuity requirement is strictly satisfied by allocating system resources based on the worst-case scenario, i.e., considering the peak frame size. Whereas, predictive admission control schemes attempt to reliably predict the future behavior of the system based upon its measured past behavior. In stead of predicting the system behaviors, pre-analyzed schemes schedule the video streams based on the pre-calculated parameters of stored video. Strict admission control schemes have the advantage of simplicity at the cost of low throughput. This can easily be understood by the example that the peak to mean ratio of coded frame sizes is 12 for the Star Wars film using the MPEG coding scheme [3]. To improve the throughput of the strict admission control schemes, predictive admission control schemes allow a given bound on the data loss/overdue rate and use measurements to estimate the expected behavior of the system. However, to collect real-time measurements incurs system overhead. In addition, the dynamics of video traffic make it very hard to guarantee the data loss/overdue rate for each video stream to be within the given bound. To overcome this shortage, the pre-analyzed schemes utilize the pre-analyzed traffic parameters of stored videos to achieve efficient and reliable video delivery. Based on some easy pre-calculated information, Lin and Hsieh [7] propose some real-time pre-analyzed admission control and buffer management algorithms for the VOD service under various service architectures.

In this paper, in addition to our previous proposed Single Segment (SS) admission control scheme [7], we propose two new schemes for pre-analyzed admission control schemes, i.e., Segmental Re-tuned (SR) and Multiple Segment (MS) admission control schemes. The SS scheme intends to improve the throughput of the strict admission control schemes by employing some easily pre-calculated information than the peak frame size from the stored video. Moreover, the SR and MS algorithms further improve the throughput of the SS scheme by divided the video into segments to increase multiplexing gain.

This paper also considers the buffer management problem in the server and the clients. For the VOD server, the buffer management algorithm can be applied either in the system design phase to calculate the minimum buffer requirement or on a real-time basis for dynamic buffer management. For the VOD clients, we propose a client buffer control algorithm which running on the server to prevent buffer overflow of the clients.

The remainder of this paper is organized as follows. In section 2, the video server architectures is presented. Section 3 describes the basic pre-analyzed scheme – the single segment (SS) scheme. Sections 4 and 5 discuss the two new schemes, the segmental re-tuned (SR) scheme and the multiple-segment (MS) scheme. Section 6 shows numerical results obtained through the computation experiments. Section 7 gives some conclusions.

2. The video server architectures

It is commonly agreed that a high-speed backbone such as ATM networks will be used to transport video streams. More importantly, it is likely that video information is stored digitally in a compressed form to save storage space and to reduce transport cost. This compression/decompression process typically requires transporting variable-bit-rate video streams, which is the case we assumed in this paper. The traffic variation of video streams makes the development of efficient and effective admission control schemes more difficult.

A key challenge involved in providing the VOD service is the need to store a huge amount of video data in an archive such that each video stream can be accessed and transmitted to the display device in real time. Various video server architectures [1,8–10,12] have been proposed, e.g., the disk array architecture, the juke box and disk array architecture, and the distributed architecture.

In a video server with the disk array architecture, video information is stored in a high-speed disk array. When a video stream is to be played, it will be first brought into a buffer (e.g., RAM) and then sent to the customer through a communication facility. In a video server with the juke box and disk array architecture, two types of storage are used. Less frequently accessed video information is stored in "juke-boxes" – larger and slower devices such as optical disks. Whereas, more frequently accessed video information is stored in high-speed disk arrays. Two cases are possible. In the first case, video streams available in the juke box but not in the disk array have to be brought into the disk array from juke box. In the second case, those video streams are brought directly to the buffer from the disks before transmission. In a video server with a distributed architecture, video information are distributed among a number of servers which cooperatively support the service.

In this paper, we consider the following four architectures and develop corresponding admission control and buffer management algorithms:

(i) Simple structure system with only one disk.

(ii) Stripping structure system with multiple disks.

(iii) Hierarchical simple structure with one juke-box and one disk.

(iv) Hierarchical stripping structure with one juke-box and multiple disks.

3. The single segment (SS) schemes

In the section, we describe the basic pre-analyzed scheme which is called Enhanced Strict Admission Control (ESAC) in [7] but renamed as Single Segment (SS) admission control in this paper.

3.1. *Mathematical models*

It is essential that video information be available at the display device by the time of its playback. As discussed previously, we refer to this as the *continuity requirement*. Assume that the disk scheduler services all active video streams in a round-robin fashion. Refer to a *cycle* as the time of such a round. Let B be the buffer size of the server in terms of data blocks. Let K_{D}, K_{J} be the block transfer rates from the disk to the buffer, and from the juke-box to the disk, respectively. Let I be the set of the new and the existing video streams, where the admission control of the new video stream(s) is to be determined. Let A_{D} and A_{J} be the worst-case disk and juke-box access overhead, respectively in one cycle. Let J_i be the maximum number of blocks can be fetched in one cycle for stream $i \in I$. The value of J_i depends on the buffer size of the client requesting stream i. For each video stream i, we pre-calculate d_{ij} which is the worst-case playback duration for stream i when exactly j blocks are fetched from the disk in each cycle. "Running window" with width j can be used to calculate d_{ij} (figure 3). Note that d_{ij} need to be calculated only once and the calculation can be done off-line by using the timestamps stored with data. It can be stored at the beginning of each video stream for admission control purposes.

Let $g_i(\cdot)$ be the function of playback duration with respect to the number of the blocks fetched in each cycle time for stream i. For the Signal Segment schemes considered in this section, $g_i(j)$ equals d_{ij}. Let τ_i be the number of blocks for stream i fetched from the disk array to the buffer in each cycle. To avoid client buffer overflow, τ_i must be less or equal to J_i. The number of blocks fetched in one cycle for each admitted stream may change with time upon call admission and/or termination.

We model the admission control problems for the four video server architectures considered as system P1, P2, P3 and P4, respectively, in the following.

With these models, the admission control is quite straight forward. That is, if and only if the corresponding system has a feasible solution, all the new video stream(s) can be admitted.

3.1.1. *Simple structure*

We formulate the admission control problem of simple structure system under SS scheme as the feasibility problem of system P1.

System P1.

$$A_{\mathrm{D}} + \sum_{k\in I} \frac{\tau_k}{K_{\mathrm{D}}} \leqslant g_i(\tau_i) \qquad \forall i \in I, \tag{3.1}$$

$$\sum_{i\in I} \tau_i \leqslant B, \tag{3.2}$$

$$\tau_i \in \{1,2,3,\ldots,J_i\} \quad \forall i \in I. \tag{3.3}$$

The left-hand-side of constraint (3.1) is the worst-case cycle time. We define this constraint as a *cycle time constraint*. Constraint (3.1) requires that no starvation occur and thus the continuity requirement is satisfied for every stream. Constraint (3.2) requires that the total amount of buffer allocated not exceed a given value B. Constraints (3.3) limit the number of block fetched in one cycle for particular stream to avoid the buffer overflow of corresponding client.

3.1.2. Stripping structure

Under this architecture, we assume that every video stream is striped into several disks and that all disks can be read simultaneously (see figure 1). Let N be the number of disks. If and only if the following system has a feasible solution, the new video stream(s) can be admitted.

System P2.

$$A_\mathrm{D} + \sum_{k \in I} \frac{\lceil \tau_k / N \rceil}{K_\mathrm{D}} \leqslant g_i(\tau_i) \quad \forall i \in I \tag{3.4}$$

and (3.2), (3.3).

System P2 is the same as system P1 except that *cycle time constraint* (constraint (3.1)) is replaced by constraint (3.4). Since the video stream data is striped into N disks (see figure 1), every disk needs to read at most $\lceil \tau_k / N \rceil$ blocks for stream i in each cycle. Other constraints have been explained previously.

3.1.3. Hierarchical simple structure

Under this architecture, video streams are stored in the juke-box and the disk (see figure 2). If the required video stream is not in the disk, we will transfer the requested video streams from the juke-box to the disk first, and then fetch it from the disk. We assume that there are two disk heads to service the data streams and can operate

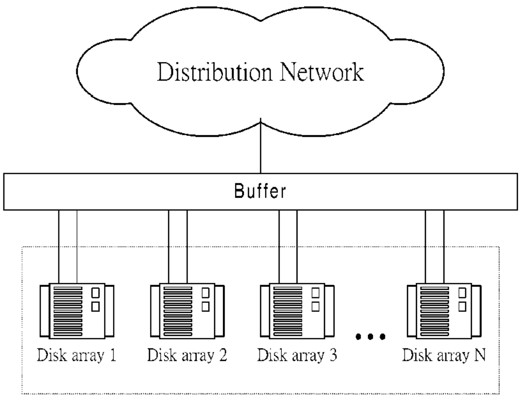


Figure 1. Stripping structure.

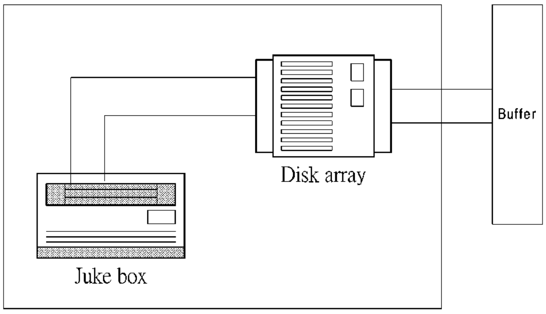


Figure 2. Hierarchical simple structure.

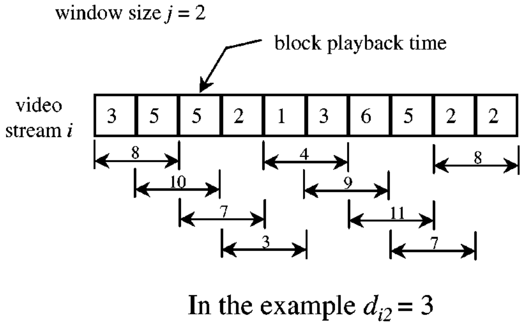


Figure 3. Calculation of d_{ij} by a running window.

simultaneously. One disk head is used to write video streams from the juke-box to the disk, while the other is used to read video streams located in the disk. Let I and I' be the set of video streams including new and existing which are originally located in the disk and juke box, respectively. If and only if the following system has a feasible solution, all new video streams can be admitted.

System P3.

$$A_{\mathrm{D}} + \sum_{k \in I \cup I'} \frac{\tau_k}{K_{\mathrm{D}}} \leqslant g_i(\tau_i) \qquad \forall i \in I \cup I', \tag{3.5}$$

$$A_{\mathrm{J}} + \sum_{k \in I'} \frac{\tau_k}{K_{\mathrm{J}}} \leqslant g_i(\tau_i) \qquad \forall i \in I \cup I', \tag{3.6}$$

$$\sum_{i \in I \cup I'} \tau_i \leqslant B, \tag{3.7}$$

$$\tau_i \in \{1, 2, 3, \ldots, J_i\} \quad \forall i \in I \cup I'. \tag{3.8}$$

The left-hand-side of constraint (3.5) is the time to access and transfer data from the disk to the buffer for video streams in I' in one cycle, while the left-hand-side of constraint (3.6) is the time to access and transfer data from the juke-box to the disk for video streams in I' in one cycle. It is clear that the larger one of the above two items is the cycle time, and therefore, constraints (3.5) and (3.6) jointly enforce the continuity requirement. Constraint (3.7) requires that the total amount of buffer allocated not exceed a given value B. Constraint (3.8) require that for each stream $i \in I \cup I'$, at most J_i number of blocks are fetched from the disk or the juke-box in one cycle.

3.1.4. *Hierarchical stripping structure*

This architecture is the same as the hierarchical simple structure except that video streams are stored in the juke-box and a number of disks, instead of a single disk. We also formulate the admission control problem as a feasibility problem of system P4 below.

System P4.

$$A_{\mathrm{D}} + \sum_{k \in I \cup I'} \frac{\tau_k/N}{K_{\mathrm{D}}} \leqslant g_i(\tau_i) \quad \forall i \in I \cup I' \tag{3.9}$$

and (3.6)–(3.8).

3.2. *Minimum buffer requirement for the server*

The following property, provides an efficient way to calculate the minimum buffer requirement of the server to admit all the streams in I for simple structure system (system P1) under SS schemes.

Property 1. Assume system P1 is infeasible under a given value of B, denoted by $\widehat{B}$. Let B^* be the minimum value of B such that system P1 is feasible. Find a solution to system P5 such that the total buffer requirement is minimized.

System P5.

$$\begin{aligned} A + \frac{\widehat{B}}{K} &\leqslant g_i(\tau_i) && \forall i \in I, \\ \tau_i &\in \{1, 2, 3, \ldots, J_i\} && \forall i \in I. \end{aligned} \tag{3.10}$$

Denote this solution by $\{\tau_i'\}$. Let B' be $\sum_{i \in I} \tau_i'$. Then, $\widehat{B} < B' \leqslant B^*$.

Proof. We first prove that $B' \leqslant B^*$. Let $\{\tau_i^*\}$ be an optimal solution associated with the minimal buffer size B^* ($B^* = \sum_{i \in I} \tau_i^*$). Then the following equation holds:

$$A + \frac{B^*}{K} \leqslant g_i(\tau_i^*) \quad \forall i \in I, \tag{3.11}$$

and for each $i \in I$, it is clear that the τ_i is set to j which is the smallest j, $j \in \{1, 2, 3, \ldots, J_i\}$, such that (3.11) is satisfied. Since $\widehat{B} < B^*$ from the assumption, for each $i \in I$, if τ_i with the smallest index j such that (3.11) is satisfied is set to j, then the calculated buffer requirement is no greater than B^*. We next prove that $\widehat{B} < B'$. Assume that $\widehat{B} \geqslant B'$. Then $\widehat{B}$ is feasible, which contradicts the assumption. This completes the proof. □

Property 1 implies that if an infeasible value of B is chosen, then the substitution of this value into system P5 will lead to a larger buffer requirement. This buffer requirement is either infeasible or optimal. Due to such strict monotonically, the buffer requirement will converge from below to the optimal value using repeated substitution.

Due to the similarity between systems P2–P4 and P1, properties similar to property 1 exist for all other structures (systems P2–P4). Based on the properties, the buffer management algorithm and admission control algorithm is developed and presented below.

Algorithm 1 (Minimum-buffer-requirement algorithm).

1. Choose a lower bound on the buffer requirement as the initial value for video streams in I and I' as the initial values for B and $B2$, respectively.[1]
2. Let
$$C = \max\left\{A_\mathrm{D} + \frac{(B + B2)}{K_\mathrm{D}}, A_\mathrm{J} + \frac{B2}{K_\mathrm{J}}\right\}.$$
3. For each $i \in I \cup I'$ find the minimum (single) $\tau_i \leqslant J_i$ such that $C \leqslant g_i(\tau_i)$. Denote the solution by $\{\tau_i'\}$.
4. Calculate $B' = \sum_{i \in I} \tau_i'$ and $B2' = \sum_{i \in I'} \tau_i'$.
5. If $B' = B$ and $B2' = B2$, then stop, the $B^* = B + B2$; otherwise, assign B to $B', B2$ to $B2'$ and go to step 2.

From the strict monotonicity implied by property 1, the buffer size strictly increases iteration by iteration if not optimal, and will converge to an optimal solution.

3.3. Admission control algorithm

The basic idea for the optimal real-time admission control algorithm is that if the minuteman requirement buffer is larger than the server's buffer, then reject the new call; otherwise, accept the call.

Algorithm 2 (Optimal-admission-control algorithm).

1. Apply Minimum-buffer-requirement algorithm to calculate the minimum buffer requirement B^*.
2. If $B^* \leqslant B$, then admit the new call(s); otherwise, reject the new call(s).

[1] I' and $B2$ are used only in hierarchical structure, i.e., the other case $B2 = 0$ and $I' = \emptyset$.

Note that to improve the performance, step 1 can be modified in such a way that if $B' > B$ at any intermediate stage of applying Minimum-buffer-requirement algorithm, then reject the call and stop.

3.4. Client buffer control algorithm

Optimal-admission-control algorithm can guarantee no starvation on the clients for video stream i if τ_i blocks are got from disk in each cycle because the total playback time of τ_i blocks is longer then the round-robin cycle time. However, excessive data will be accumulate in the buffers of the client for the following cycles. Without proper buffer control, the accumulated of excessive data will eventually overflow the buffers. Thus, we design the client buffer control algorithm running on the server to limit the excessive data on each client.

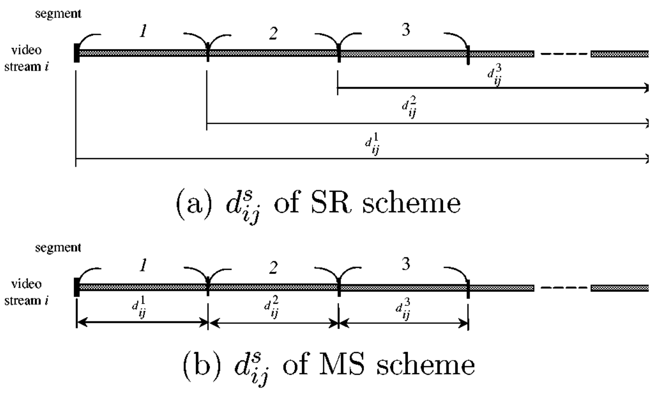
Let the *Cycle_Time* $= A_{\mathrm{D}} + \sum_{k \in I}(\tau_k/K_{\mathrm{D}})$, obtained from Optimal-admission-control algorithm, be the disk scheduler‘s round-robin cycle time. Let T_l^i be the playback time of lth block in stream i. This block playback time can easily obtained in pre-analyzed phase. Let T^i_{remain} be the excessive playback time of the data remain in the client buffer after a cycle and G^i be the actual number of blocks fetched from disk in this cycle for stream i. Then, the algorithm is presented below:

Algorithm 3 (Client-buffer-control algorithm).
For each cycle:

1. At the beginning of each cycle, if there is a new stream (say, stream k) admitted in the cycle, $T^k_{\text{remain}} \leftarrow 0$, $l^k \leftarrow 0$, $I \leftarrow I \cup \{k\}$, *Cycle_Time* $\leftarrow A_{\mathrm{D}} + \sum_{j \in I}(\tau_j/K_{\mathrm{D}})$.
2. If there is a termination stream (say stream k'), then update the length of round-robin cycle: $I \leftarrow I - \{k'\}$, *Cycle_Time* $\leftarrow A_{\mathrm{D}} + \sum_{j \in I}(\tau_j/K_{\mathrm{D}})$.
3. Find the minimum G^i, such that $T^i_{\text{remain}} + \sum_{k=l^i}^{l^i+G^i} T^i_k >$ *Cycle_Time*, $\forall i \in I$, $G^i \in \{1, 2, 3, \ldots J_i\}$.
4. The server fetch G^i (from l^ith to $(l^i + G^i)$th) blocks for stream i, $\forall i \in I$, and transmits these data to corresponding clients during the cycle.
5. $T^i_{\text{remain}} \leftarrow T^i_{\text{remain}} + \sum_{k=l^i}^{l^i+G^i} T^i_k -$ *Cycle_Time*.
   $l^i \leftarrow l^i + G^i$, $\forall i \in I$.

4. The Segmental Re-tuned (SR) scheme

In Single Segment (SS) scheme, whether the new user(s) will be admitted into the system is based on examining d_{ij} values of the new and the admitted users. Since the value of d_{ij} is got by scanning the video from the beginning to the end, it may be too conservative to use the d_{ij}'s of the existing admitted users in the system for admission control. The basic concept of the SR scheme is to update these d_{ij}'s before



(a) d_{ij}^s of SR scheme

(b) d_{ij}^s of MS scheme

Figure 4. Difference on the definition of d_{ij}^s between SR and MS schemes.

proceeding admission control. When new user(s) come, the minimum playback time for the admitted video should be rescanned starting from the current time instead of from the beginning to the end. Note that d_{ij} of stream i is monotonic increasing with the starting time of scanning. Since the SR scheme re-tunes d_{ij} to a larger value when a new request arrives, it has performance at least as good as, if not better than, the performance of SS scheme. The major problem of this re-tuned scheme is that it is very hard to update d_{ij} real-time by scanning video trace on-line.

To overcome above difficulty, under the SR scheme, every video is divided into segments according to a fixed playback time T. The minimum playback time at segment s of the stream i, d_{ij}^s, is calculated from the beginning of segment s to the end of the video. Figure 4(a) is an illustration for the d_{ij}^s. Since every video stream is stored in storage devices, d_{ij}^s can be easily pre-calculated and pre-stored. We can observe that d_{ij}^s is a monotonically increasing function of s.

The system time of SR schemes is also divided into fixed time periods. The length of a period is the same as playing time of a segment. When a new user(s) needs to be admitted, the system first examines which segment the admitted users are currently in and then, uses the admitted streams' current d_{ij}^s's and new user d_{ij}^1's for admission decisions. For example, in figure 5, a new user requests to enter the system at some point in period x. At the same time, there have been users 1 and 2 already in the system. Both users 1 and 2 have two segments in the period. They are denoted as a and $a+1$ for user 1 and b and $b+1$ for user 2, respectively. The SS scheme takes d_{1j}^1, d_{2j}^1 and d_{3j}^1 as the parameters to determine whether to admit user 3. However, the SR scheme uses the three data, d_{1j}^a, d_{2j}^b and d_{3j}^1, to determine whether the new user, user 3, can be admitted into the system. Since d_{ij}^s is a monotonically increasing function of s, the SR scheme performs at least as good as the SS scheme.

By redefining the g_i function to the corresponding d_{ij}^s in the period the new user(s) arrived. Algorithms for the SS schemes in previous section are ready for the SR scheme.

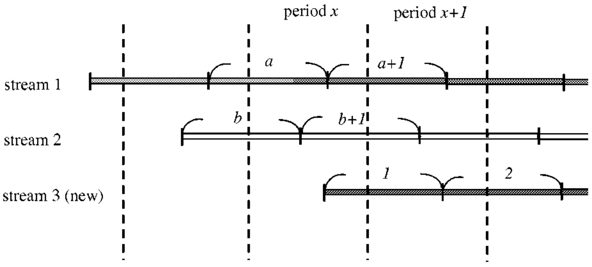


Figure 5. An illustration for the relation between system periods and video segmentations.

5. The Multiple-Segment (MS) scheme

Just like the SR scheme, the video streams are also divided into equal playback time segments under the MS scheme. However, unlike the SR scheme which calculates d_{ij}^s by scanning from the beginning of the segment s to the end of the video, the d_{ij}^s of the MS scheme is obtained by scanning the video from the beginning to the end of the same segment, i.e., d_{ij}^s is effective only within one segment (figure 4(b)).

In the MS scheme, the time is also divided into fixed periods. Note that in our system the period and the video segment do not have to be aligned. A new user can be admitted to the system only if every segment of the new video can be admitted to the system. Since the d_{ij}^s is now effective only in one segment, all periods which the new video(s) lasting is needed to be examined to respect the *continuity requirement.* Figure 5 depicts how to admit a new user(s). In this example, to admit user 3 arrived in period x, we have to admit all segments of the requested video. To admitted the first segment, we have to admit it both in period x and period $x+1$. Therefore, we take $\min\{d_{1j}^a, d_{1j}^{a+1}\}$, $\min\{d_{2j}^b, d_{2j}^{b+1}\}$ and d_{3j}^1 into consideration for segment 1. Since the second segment across periods $x+1$ and $x+2$, we have to consider $\min\{d_{1j}^{a+1}, d_{1j}^{a+2}\}$, $\min\{d_{2j}^{b+1}, d_{2j}^{b+2}\}$ and $\min\{d_{3j}^1, d_{3j}^2\}$. Assume user 3 lasts total m periods, then user 3 can be admitted only if all the m periods can admit user 3; otherwise, user 3 will be rejected.

Since both the video and the system time are divided into segments or periods and each segments is across at most two periods. the mutual relationship between segments is limited within two periods. From this point of view, the segmentation effect of the MS scheme provides some kind of horizontal multiplexing gain between video streams by limiting the correlation between periods.

6. Experimental results

In order to evaluate and compare the performance of the proposed admission control algorithms, computational experiments are performed.

We carried out simulations using traces of real video sources. The data set Starwars consists of frame sizes of the film Starwars encoded in MPEG-1. It has mean rate

Table 1
System parameters.

Disk transfer rate	6 MB/s
Disk worst-case overhead for each stream	25 ms
Block size	50 KB
Stripping degree	4
Juke-box transfer rate	600 KB/s
Juke-box worst-case overhead for each stream	250 ms
Server buffer size	128 MB
Client buffer size	2 MB
Video length	120.23 min
System period	10 min
Video title	starwars
Video coding scheme	MPEG

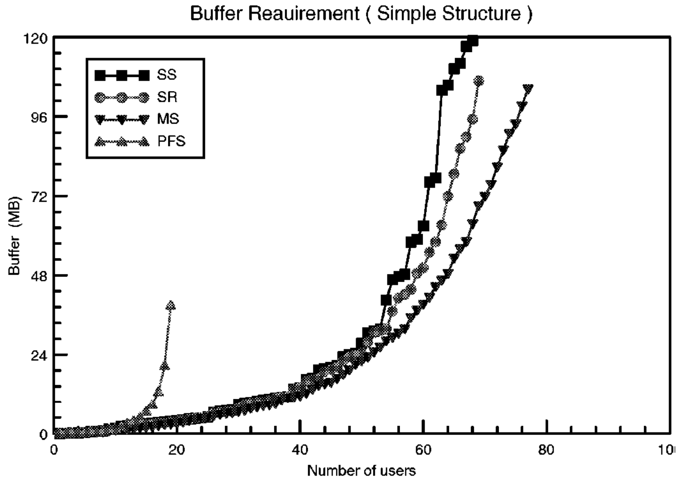


Figure 6. Buffer requirement for simple structure.

of 374.4 kbit/s and a peak rate (largest frame divided by frame period) of 4.46 kbit/s. Unless otherwise mentioned, the system parameters used in the computational experiments are given in table 1.

Figure 6 shows the buffer requirement for different number of admissible users in simple structure. It is observed that the SR and MS schemes achieve 250% and 300% improvement respectively on the maximum number of admissible users over the peak frame size (PFS) scheme for fixed amount of server buffer, say, 30 MB. Similar situations happen on the stripping structure and hierarchical simple structure (figures 7 and 8). In addition, these figures give us an idea of the amount of buffer required on the server. The buffer requirement increases nearly linearly with the number of admissible users when the users is under 50 on the SR or MS scheme. However, if the number of users is more then 50, the buffer requirement increases dramatically. This means that the disk bandwidth becomes bottleneck as the number of users increasing.

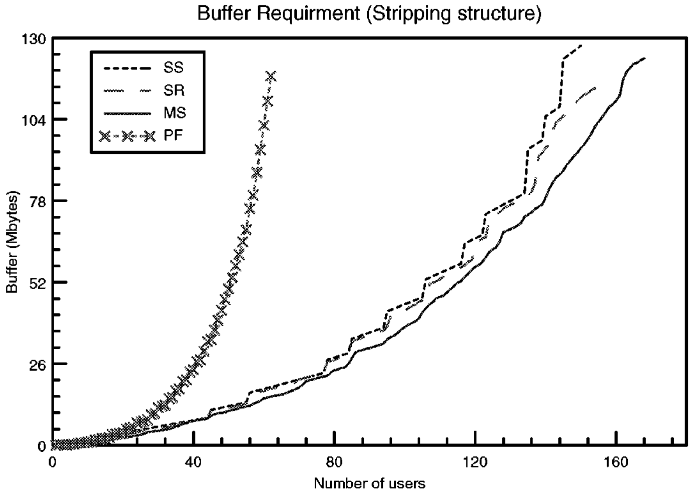


Figure 7. Buffer requirement for stripping structure.

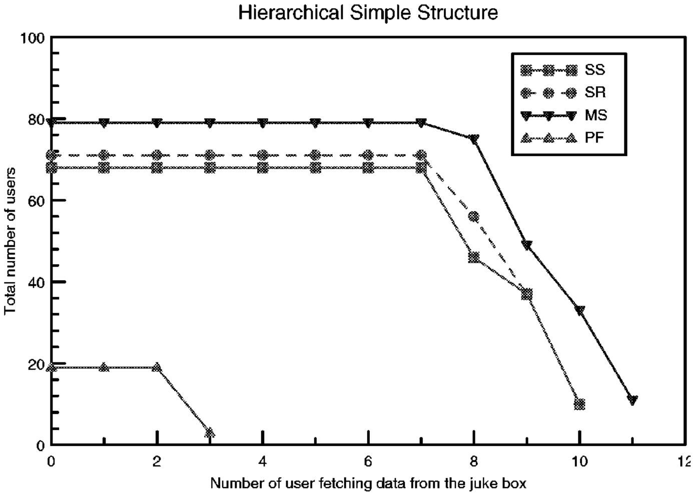


Figure 8. The number of admissible users for hierarchical simple structure.

The disk stripping can increase the disk bandwidth. For example, the stripping degree equals 4 in our case, so the disk bandwidth is 4 times those in the simple structure. Comparing figure 6 to figure 7, we observe that the gain on the maximum system throughput by using disk stripping is about 2 rather than 4.

Figures 8 and 9 show that the maximum number of admissible users is kept the same when the number of juke-box users is limited, and decreases rapidly when the number of juke-box users reaches a small threshold. This observation suggests that in view of the maximum system throughput, the number of admissible juke-box users should be limited.

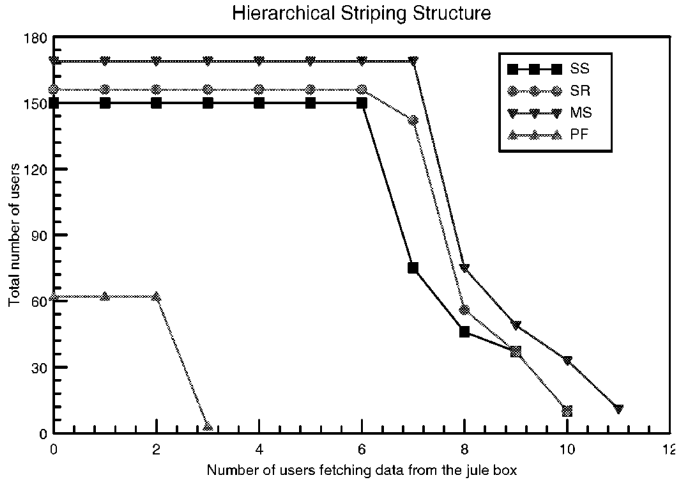


Figure 9. The number of admissible users for hierarchical stripping structure.

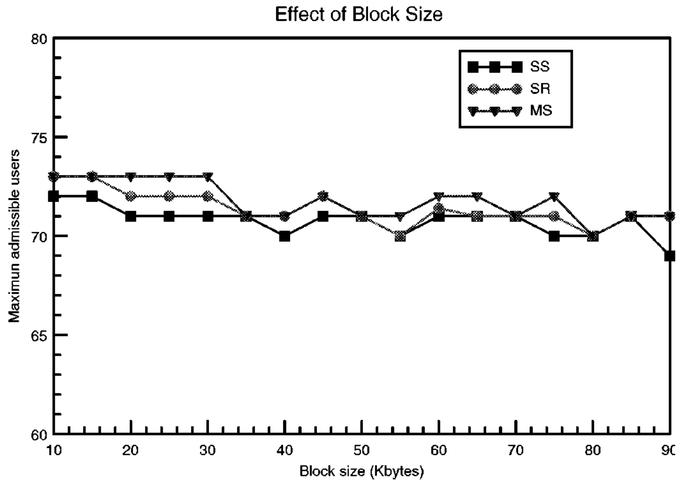


Figure 10. The effect of the total number of admissible users using different block sizes.

Figure 10 shows the effect of data block size. The increasing of block size only slightly decreases the maximum admissible users, i.e., the system performance is not sensitive to the block size. In fact, larger block size may prefer in designing a system because it can reduce the operation complexity.

Figure 11 shows that the effect of the server buffer size of maximum admissible users. As we expected, the maximum number of admissible user increases as the buffer size of the server increasing. The increasing seems to be linear when buffer size is small but saturates as buffer become large. In these examples, MS scheme can admits at most 108 users; SS and SR scheme at most 100 users, but PS scheme can admit only 24 users.

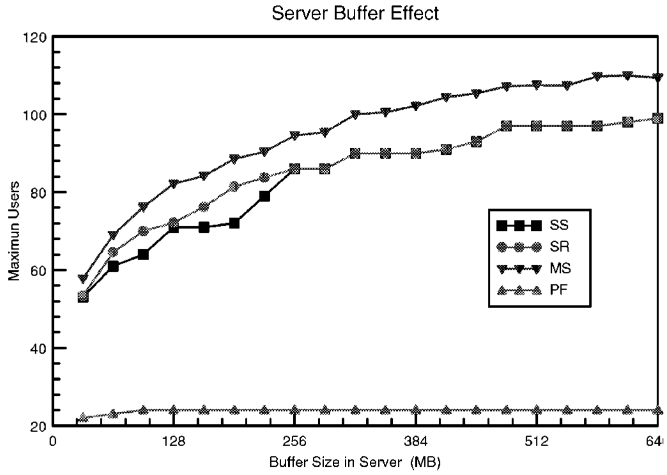


Figure 11. The effect of the server buffer size to the number of admissible users.

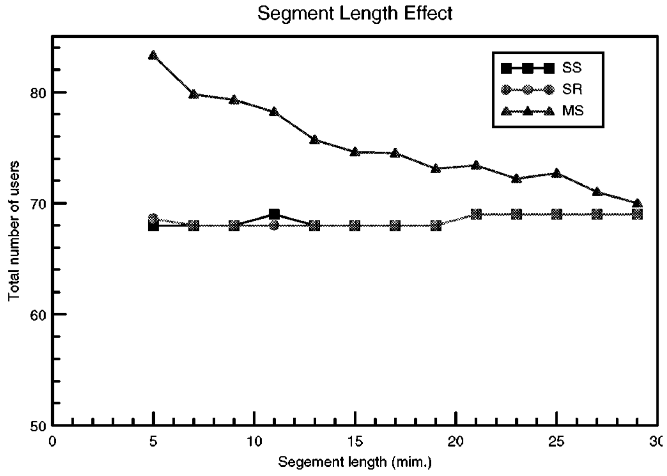


Figure 12. The effect of the total number of admissible users using different length of segment playback time.

Figure 12 is the relationship between the length of periods and maximum throughput. As we expect, more users can be admitted when smaller period used. The MS scheme outperforms SS scheme 5, 18 and 25% for time period equal to 30, 10 and 5 min, respectively, in this experiments. Segmentation deed increases the throughput by providing the horizontal multiplexing gains between the video streams.

Assume the stream request is Poisson process with the mean λ (requests/hour) under a system with the simple structure and parameters given in table 1. A request is blocked if the requested stream can not be admitted at the time the request arrived. Figure 13 shows the blocking probability with respect to different offer load for dif-

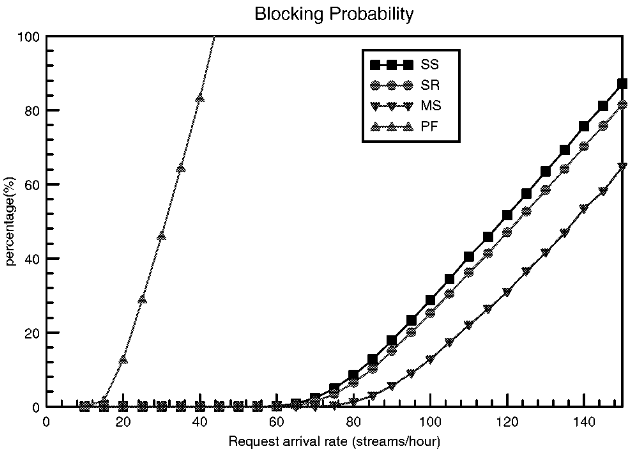


Figure 13. Request blocking probability versus request rate.

ferent schemes. This figure can be used to estimate customer loss (revenue loss) of different schemes under different system load. Again, the MS scheme has the best performance.

7. Conclusions

In this paper, we propose two segmental based schemes for admission control and buffer management algorithms of VOD systems under various system architectures. By pre-analyzing the characteristics of the stored video, the new proposed segmentation schemes deed improve the system throughput. The computational experiments shows that the Segmental Returned scheme and the Multiple Segment scheme can achieve over 175% and over 200% improvement, respectively, comparing with conventional PFS scheme in the system throughput. Depending on the playback time of a segment, Multiple Segment scheme out performs 5–25% better than the Segmental Re-tuned and the Single Segment schemes.

References

[1] D.P. Anderson, Y. Osawa and R. Govindan, A file system for continuous media, ACM Transactions on Computer Systems 11(2) (1993).

[2] D. Clark, S. Shenker and L. Zhang, Supporting real-time applications in an integrated services network: Architecture and mechanism, in: *Proc. of the ACM SIGCOMM* (1992) pp. 14–26.

[3] M.W. Garrett, Contributions towards real-time services on packet-switched networks, Ph.D. thesis, Columbia University (March 1993).

[4] S. Jamin, S. Shenker, L. Zhang and D.D. Clark, An admission control algorithm for predictive real-time service, in: *Third Internat. Workshop on Network and Operating System Support for Digital Audio and Video* (November 1992) pp. 349–356.

[5] D. Le Gall, A video compression standard for multimedia applications, Communications of the ACM 34(4) (1991) 47–58.

[6] F.Y.-S. Lin, Optimal real-time admission control for the Video-On-Demand (VOD) service, IEEE Transactions on Broadcasting 44(4) (1998) 402–408.

[7] F.Y.-S. Lin and K.-T. Hsieh, Real-time admission control and buffer management algorithms for the Video-On-Demand (VOD) service under various video server architectures, in: *Proc. of 11th Internat. Conf. on Information Networking (ICOIN-11)* (January 1997).

[8] P. Lougher and D. Shepherd, The design and implementation of a continuous media storage server, in: *Third Internat. Workshop on Network and Operating System Support for Digital Audio and Video* (November 1992) pp. 69–80.

[9] P.V. Rangan and H.M. Vin, Designing file systems for digital video and audio, in: *Proc. of the 13th ACM Symposium on Operating Systems Principles* (1991) pp. 81–94.

[10] P.V. Rangan, H.M. Vin and S. Ramanathan, Designing an on-demand multimedia service, IEEE Communications Magazine 30(7) (1992) 56–65.

[11] W.D. Sincoskie, Video on Demand: Is it feasible? in: *Proc. of the IEEE GLOBECOM'90*, San Diego, CA (1990).

[12] L. Vaitzblit, The design and implementation of a high bandwidth file service for continuous media, MIT thesis (September 1991).

[13] H.M. Vin and P.V. Rangan, Admission control algorithms for multimedia on-demand servers, in: *Third Internat. Workshop on Network and Operating System Support for Digital Audio and Video* (November 1992) pp. 56–68.