Journal of the Operational Research Society (2001) 52, 55–63

www.palgrave.com/jors

An incentive charging scheme for video-on-demand

Y-W Leung[1]* and EWM Wong[2]

[1]*Hong Kong Baptist University, and* [2]*City University of Hong Kong, Hong Kong*

Video-on-demand (VOD) systems can provide either an *individual service* or *batch service.* For individual service, a user can receive video immediately after making a request and he/she can perform interactive operations (such as pause, jump, fast forward and rewind), and the system uses one video stream to serve one user. For batch service, a user has to wait after making a request and cannot perform interactive operations, but the system can use one video stream to serve a batch of users. Therefore, individual service has a better quality while batch service requires less resources to serve each user. In this paper, we consider a VOD system providing both services and propose an *incentive charging scheme* to optimize the coexistence of both services. This scheme imposes a lower service charge on batch service in order to attract users to choose this service. Consequently, the service provider can get more revenue by serving more concurrent users via batch service and users can choose their preferred services. We analyze the incentive charging scheme and maximize the mean revenue subject to a given availability specification. The numerical results show that the incentive charging scheme is particularly effective in peak hours when the demand for the VOD service is large.

Keywords: information systems; telecommunications; stochastic optimization

Introduction

A *video-on-demand* (VOD) system provides an electronic video rental service to geographically distributed users.[1,2] Using this service, users can select and watch video programmes at a convenient time and place, and they may interact with the programmes using interactive operations such as pause, jump, fast forward and rewind.

Figure 1 shows a typical VOD system. A *server system* stores a collection of video programmes, and it delivers video to the users through an information network upon their requests. To design a VOD system, two problems must be addressed: (1) how to deliver video from the server system to the users, and (2) how to design a server system.

We first consider the problem of delivering video. The server system can deliver video through an existing information network, and then through the existing cables between the local exchanges and the users using digital subscriber line methods.[3,4] For example, the following subscriber line methods are used in the first commercial VOD system (called *iTV system*) in the world:[5–7]

- When there is an optical fibre between a local exchange and a building, video is delivered from the local exchange to the building through the optical fibre and then it is delivered to the user through the telephone twisted pairs within the building using *Very High Speed Digital Subscriber Line (VDSL).*[3] VDSL is relatively cheap and it is suitable for short-distance communication. VDSL is used for most of the users in the iTV system.
- When there are only twisted pairs between a local exchange and the users, video is delivered from the local exchange to the user through the twisted pairs using *Asymmetric Digital Subscriber Line (ADSL).*[4] ADSL is relatively expensive, but it is suitable for longer-distance communication. As ADSL becomes more popular, it will be less expensive because of economy of scale and competition.

We now consider the problem of designing server systems. The simplest server system is composed of one server computer. However, video has a high playout rate, and therefore a server computer can only provide a limited number of video streams. For example, a typical server computer can provide about 50 streams of MPEG video at 1.5 Mbps/stream. If the system is used for large-scale applications, it has to serve many concurrent users (eg 10 000). For this purpose, the server system can use a large number of server computers, but the resulting cost and maintenance overhead are high. Alternatively, the system can provide a service that has a lower quality but requires less resources (eg input/output bandwidth) to serve each user, so that the system can serve more concurrent users. There are two services having different qualities and resource requirements:

- *Individual service.* Using individual service, a user can receive video immediately after making a request. He can perform interactive operations (such as pause, rewind and fast forward) to control the presentation of the video. For

**Correspondence: Y-W Leung, Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.*
Email: ywleung@comp.hkbu.edu.hk

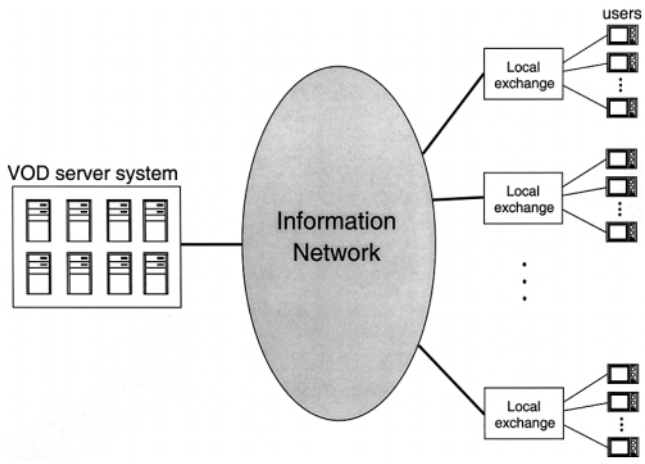


Figure 1 A typical VOD system.

example, he can pause a programme temporarily when the telephone rings, he can perform a rewind operation to watch the exciting part of a movie again, he can perform a fast forward operation to skip the international news and watch the entertainment news, etc. The server system uses one video stream to serve one user.

- *Batch service.* Using batch service,[8–13] a user has to wait after making a request until the system collects a batch of requests for the same video programme. After commencement, he receives video but he cannot perform interactive operations[8–10] (or can only perform some restricted operations at the expenses of increasing the system complexity and using more resources[11–13]). The server system uses one video stream to serve a batch of users. Compared with individual service, batch service has a lower quality but it requires less resources to serve each user.

In the service industry, a service provider may provide services with different qualities and different service charges in order to fulfill the needs of different users.[14] For example, railways offer express and regular freight services, and the post office offers priority and regular mail services. In a similar manner, a VOD system can provide both individual service and batch service at different service charges. A user who prefers a good quality of service can choose the individual service, and a user who prefers a low service charge can choose the batch service. Since these services require different amount of resources per user, it is necessary to design and optimize a charging scheme in order to optimize their coexistence.

In this paper, we consider a VOD system providing both services and propose an *incentive charging scheme* for it. This scheme imposes a lower service charge on the batch service in order to attract the users to choose this lower-quality service. We analyze the incentive charging scheme and maximize the mean revenue, subject to a given availability specification. We demonstrate that this scheme benefits not only the service provider (with higher revenue) but also the users (with higher satisfaction due to higher availability and more service choices). In addition, the results provide some interesting insights for the design and optimization of VOD systems.

Incentive charging scheme

The VOD system provides M video programmes which may have different popularities and duration (eg they can be movies, news, video magazines, karaoke video, etc.). It provides both individual service and batch service. If a user wants to start to watch a video programme immediately or he wants to interact with it, he can choose the individual service. On the other hand, if another user does not mind to wait some minutes before commencement and he just wants to watch a video programme passively, he can choose the batch service to enjoy a lower service charge.

For individual service, the system allocates one dedicated video stream to serve each user. For batch service, when the system receives an initial request for a video programme, it waits W min (eg 10 min) to collect more requests for the same programme, and then allocates one video stream to serve this batch of users. This guarantees that the waiting time for batch service is at most W min.

From the users' point of view, individual service is better than batch service. The users must be given an incentive to choose the batch service; otherwise, no one would be willing to choose it. For this purpose, the incentive charging scheme imposes a lower service charge on the batch service.

If the individual service is chosen for the ith video programme, the service charge is C_i, where C_i depends on the popularity and duration of this programme. If the batch service is chosen for the same programme, the service charge is δC_i, where $0 < \delta \leqslant 1$ and δ is called the *relative service charge* (eg $\delta = 0.90$ means a 10% discount). In other words, the service charge of the batch service depends on a discount factor as well as the popularity and duration of the video programme. δ is a design parameter and we will determine δ in the next section.

When the system is serving many ongoing users, it may not have sufficient free resources to serve any additional user. In this case, service is not available to new users until at least an ongoing user has terminated his VOD session. We measure the availability of a service in term of *blocking probability* which is the probability that this service is not available because there is no free video stream, so that the availability is equal to one minus the blocking probability. The incentive charging scheme imposes the following availability specification for each service. Let $\Im_{\mathrm{I}}$ and $\Im_{\mathrm{B}}$ be the blocking probabilities of the individual service and the batch service, respectively. The scheme ensures $\Im_{\mathrm{I}} \leqslant \Im_{\mathrm{I}}^*$ and $\Im_{\mathrm{B}} \leqslant \Im_{\mathrm{B}}^*$ where $\Im_{\mathrm{I}}^*$ and $\Im_{\mathrm{B}}^*$ are the availability requirements specified by the service provider.

Remark 1. The incentive charging scheme is advantageous to both the service provider and the users:

- The system can serve more concurrent users via the batch service. Therefore, the service provider can get more revenue and provide a higher availability to the users.
- The users can choose their preferred services, and they can experience a higher availability.

Remark 2. We note that the batch service may not be very effective for the non-popular video programmes, because it is likely that there are only a small number of users in a batch. Nevertheless, our scheme provides the batch service at a fixed discount for every programme for two reasons: (1) the scheme is simple and easily understandable to the general public, and (2) the demand for the non-popular programmes is relatively small. However, the scheme and the subsequent analysis can easily be generalized to the case in which the batch service is only provided for the popular programmes. The details of this generalization are included in a fuller version of this paper.[15]

Remark 3. The incentive charging scheme follows the practice adopted in the cinemas and the video rental stores, in which the service charge for watching a particular video programme is fixed even if the user does not watch the entire programme.

Modelling, analysis and optimization

In this section, we formulate an analytic model for the incentive charging scheme. We analyze the mean revenue and the blocking probabilities, and then maximize the mean revenue subject to the given availability specification.

Modelling

The VOD system provides N video streams, where a video stream requires resources in the server (eg I/O bandwidth and buffer) and the network (eg communication channel). Among these streams, N_B streams and N_I streams are allocated for the batch service and the individual service, respectively, so that the availability of one service will not be affected by the other service. By choosing suitable values for N_B and N_I we can ensure $\Im_I \leqslant \Im_I^*$ and $\Im_B \leqslant \Im_B^*$.

We let α_i be the probability that a request chooses the ith video programme (where $\alpha_1 \geqslant \alpha_2 \geqslant \ldots \geqslant \alpha_M$ without loss of generality). We let τ_i be the duration of this programme. If a user chooses the individual service for this programme, he may perform interactive operations and hence the duration is a random variable. We let the mean duration be $\overline{\tau}_i$.

We let β be the probability that a user chooses the batch service, where β depends on δ (ie a choice between individual service and batch service is primarily determined by the relative service charge). If the relative service charge δ is smaller (ie the discount is larger), more users are attracted to choose the batch service and hence β is larger. We expect that β is sensitive to δ in practice, because entertainment service is optional and hence its demand is sensitive to its service charge. The exact relationship between β and δ is a topic of marketing survey and is outside the scope of this study. In our study, we assume that this relationship is given.

The users make requests to the system to initiate new VOD sessions. The arrival of these requests follow a Poisson process with rate λ, where λ is fixed over a particular period (eg the peak hours).[9,10,12] This Poisson model is suitable because the VOD service should be provided for a large number of potential users in order to justify its investment cost. For example, the iTV system[7] has more than 80 000 family subscribers or 320 000 potential users for an average of four members per family.[16]

We denote a Poisson process with rate λ by $Poisson(\lambda)$. In the following, we will apply the following property of Poisson process. If $Poisson(\lambda)$ has two types of arrivals and the probability that an arrival is type-1 is p, then $Poisson(\lambda)$ can be decomposed into two independent Poisson processes $Poisson(\lambda p)$ and $Poisson(\lambda(1-p))$. These two processes model the type-1 and type-2 arrivals, respectively. The proof for this property can be found in ref. 17.

$Poisson(\lambda)$ models the arrivals of requests for VOD sessions. We decompose $Poisson(\lambda)$ as follows:

- Each arrival of $Poisson(\lambda)$ can choose one of the M video programmes. We decompose $Poisson(\lambda)$ into M independent Poisson processes $Poisson(\lambda\alpha_1)$, $Poisson(\lambda\alpha_2)$, ..., $Poisson(\lambda\alpha_M)$, such that $Poisson(\lambda\alpha_i)$ models the arrivals of requests for the ith programme.
- Each arrival of $Poisson(\lambda\alpha_i)$ can choose either the batch service or the individual service. We decompose $Poisson(\lambda\alpha_i)$ into two independent Poisson process $Poisson(\lambda\alpha_i\beta)$ and $Poisson(\lambda\alpha_i(1-\beta))$, such that they model the arrivals of requests for the ith programme choosing the batch service and the individual service respectively.

The overall decomposition of $Poisson(\lambda)$ is shown in Figure 2.

Analysis of individual service

The following M processes model the arrivals of requests for the individual service: $Poisson(\lambda\alpha_1(1-\beta))$, $Poisson(\lambda\alpha_2(1-\beta))$, ..., $Poisson(\lambda\alpha_M(1-\beta))$. The requests from different arrival processes require different service times and attract different service charges. In this subsection, we derive the availability and the mean revenue of the individual service in the form of closed form expressions.

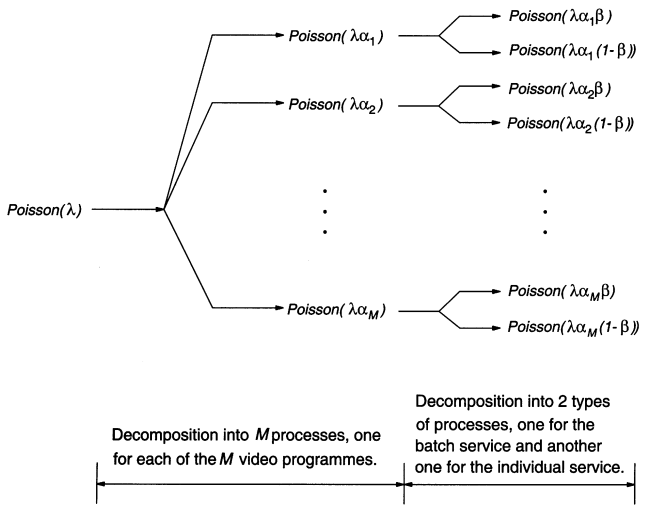


Figure 2 Decomposition of the arrival process.

We aggregate the M arrival processes into one process. The total arrival rate λ_I of this process is given by

$$\lambda_I = \sum_{i=1}^{M} \lambda\alpha_i(1-\beta) = \lambda(1-\beta). \tag{1}$$

The M arrival processes have different service times. After aggregation, the mean service time of a session $\overline{\tau}_I$ is equal to the weighted mean service time of the original M processes. $\overline{\tau}_I$ is given by

$$\overline{\tau}_I = \sum_{i=1}^{M} \left(\frac{\lambda\alpha_i(1-\beta)}{\sum_{j=1}^{M} \lambda\alpha_j(1-\beta)} \right) \overline{\tau}_i = \sum_{i=1}^{M} \alpha_i \overline{\tau}_i. \tag{2}$$

The system provides N_I video streams for the individual service. It can be shown that the system can be modeled as an MGN_IN_I queue. For this queueing system, the probability that there are n ongoing users is given by[18,19]

$$\pi_I(n) = \frac{\frac{1}{n!}(\lambda_I\overline{\tau}_I)^n}{\sum_{j=0}^{N_I} \frac{1}{j!}(\lambda_I\overline{\tau}_I)^j} \tag{3}$$

When there are N_I ongoing users, all the N_I video streams are being occupied, and consequently any new request will be blocked. The blocking probability $\Im_I$ is therefore given by

$$\Im_I = \pi_I(N_I) = \frac{\frac{1}{N_I!}(\lambda_I\overline{\tau}_I)^{N_I}}{\sum_{j=0}^{N_I} \frac{1}{j!}(\lambda_I\overline{\tau}_I)^j}. \tag{4}$$

The M video programmes have different service charges. After aggregation, the mean service charge for any video programme C_I is equal to the weighted mean service charges of the M video programmes. C_I is given by

$$C_I = \sum_{i=1}^{M} \left(\frac{\lambda\alpha_i(1-\beta)}{\sum_{j=1}^{M} \lambda\alpha_j(1-\beta)} \right) C_i = \sum_{i=1}^{M} \alpha_i C_i \tag{5}$$

When there is one ongoing user, the mean revenue per unit time R_I that the system can get is given by

$$R_I = \frac{C_I}{\overline{\tau}_I} = \frac{\sum_{i=1}^{M} \alpha_i C_i}{\sum_{i=1}^{M} \alpha_i \overline{\tau}_i}. \tag{6}$$

The mean revenue per unit time $\Re_I$ can be determined as follows:

$$\begin{aligned} \Re_I &= \sum_{n=1}^{N_I} (n\, R_I)\, \pi_I(n) \\ &= R_I \left(\frac{(\lambda_I\overline{\tau}_I) \sum_{n=1}^{N_I} \frac{1}{(n-1)!}(\lambda_I\overline{\tau}_I)^{n-1}}{\sum_{j=0}^{N_I} \frac{1}{j!}(\lambda_I\overline{\tau}_I)^j} \right) \\ &= \lambda_I C_I (1-\Im_I) \end{aligned} \tag{7}$$

Analysis of batch service

The following M processes model the arrivals of requests for the batch service: $Poisson(\lambda\alpha_1\beta)$, $Poisson(\lambda\alpha_2\beta), \ldots,$ $Poisson(\lambda\alpha_M\beta)$. The requests from different arrival processes require different service times. In this subsection, we derive the availability and the mean revenue of the batch service in the form of closed form expressions.

Consider the ith video programme. The process $Poisson(\lambda\alpha_i\beta)$ models the arrival of requests for this programme. The system allocates only one video stream to serve a batch of requests. From the system's point of view, a batch of arrivals can be regarded as an *effective* arrival, such that each effective arrival requires one video stream. We determine this effective arrival rate as follows. We note that the interarrival time of a Poisson process is exponentially distributed, and the exponential distribution has the memoryless property (ie if X is exponentially distributed, then $P(X > s+t \mid X > t) = P(X > s)$).[17] After the system allocates a video stream to a batch of requests, the mean time before there will be a new request for the same programme is equal to the mean interarrival time of $Poisson(\lambda\alpha_i\beta)$ (ie $1/\lambda\alpha_i\beta$). Therefore, the mean time between two successive allocations of video streams is

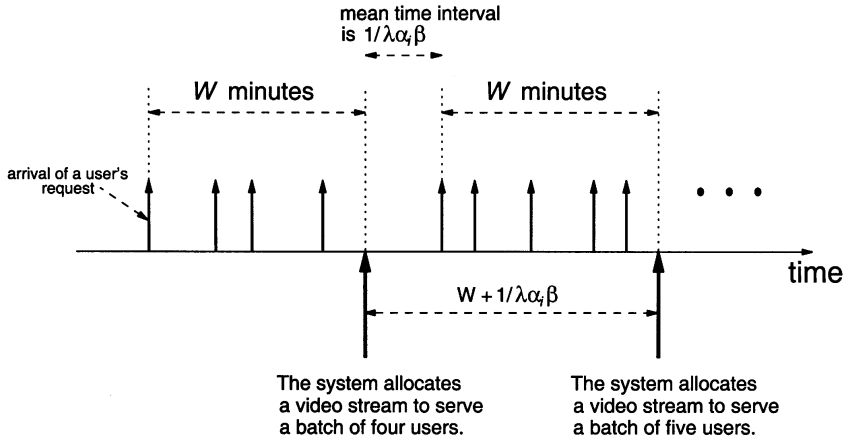


Figure 3 An example to illustrate the mean time between two successive allocations of video streams in the batch service.

$W + 1/\lambda\alpha_i\beta$ (see Figure 3). Then the effective arrival rate λ_i' is given by

$$\lambda_i' = \left(W + \frac{1}{\lambda\alpha_i\beta}\right)^{-1} = \frac{\lambda\alpha_i\beta}{(1 + \lambda\alpha_i\beta W)}. \tag{8}$$

Since the number of video programmes M is not small in practice (eg the iTV system provides more than 250 video programmes[7]) and W is small compared with the duration of the video programmes, the interarrival time of batch arrivals can be considered to be exponentially distributed. We analyze the batch service in a manner similar to that for the individual service.

We aggregate the M arrival processes into one process. The total arrival rate λ_B of this process is given by

$$\lambda_B = \sum_{i=1}^{M} \lambda_i' = \sum_{i=1}^{M} \frac{\lambda\alpha_i\beta}{1 + \lambda\alpha_i\beta W} \tag{9}$$

Using the batch service, a user cannot perform interactive operations, and therefore the duration for watching the ith video programme is fixed at τ_i. After aggregation, the mean service time of a session $\overline{\tau}_B$ is equal to the weighted average of the durations of the M video programmes and $\overline{\tau}_B$ is given by

$$\overline{\tau}_B = \sum_{i=1}^{M} \left(\frac{\lambda_i'}{\sum_{j=1}^{M} \lambda_j'} \right) \tau_i = \frac{\sum_{i=1}^{M} \lambda_i'\tau_i}{\sum_{j=1}^{M} \lambda_j'}. \tag{10}$$

The system provides N_B video streams for the batch service, and therefore there can be n batches of users in the system where $n = 0, 1, 2, \ldots, N_B$. It can be shown that the system can be modeled as an MGN_BN_B queue. For this queueing system, the probability that there are n ongoing batches of users is given by[18,19]

$$\pi_B(n) = \frac{\frac{1}{n!}(\lambda_B\overline{\tau}_B)^n}{\sum_{j=0}^{N_B} \frac{1}{j!}(\lambda_B\overline{\tau}_B)^j} \tag{11}$$

When there are N_B ongoing batches of users, all the N_B video streams are being occupied, and consequently any new batch will be blocked. The blocking probability is given by

$$\Im_B = \pi_B(N_B) = \frac{\frac{1}{N_B!}(\lambda_B\overline{\tau}_B)^{N_B}}{\sum_{j=0}^{N_B} \frac{1}{j!}(\lambda_B\overline{\tau}_B)^j} \tag{12}$$

For the ith video programme, the service charge is δC_i and the mean number of users in a batch is $(1 + \lambda\alpha_i\beta W)$. When there is one batch of users for the ith video programme, the mean revenue that the system can get is $\delta C_i(1 + \lambda\alpha_i\beta W)$. After aggregation, when there is one batch of users, the mean revenue is given by the following weighted average:

$$C_B = \sum_{i=1}^{M} \left(\frac{\lambda_i'}{\sum_{j=1}^{M} \lambda_j'} \right) \delta\, C_i(1 + \lambda\alpha_i\beta W) \tag{13}$$

and the mean revenue per unit time R_B for one batch of users is given by

$$R_B = \frac{C_B}{\overline{\tau}_B} = \frac{\sum_{i=1}^{M} \lambda_i'\delta C_i(1 + \lambda\alpha_i\beta W)}{\sum_{i=1}^{M} \lambda_i'\tau_i}. \tag{14}$$

Then the mean revenue per unit time $\Re_B$ can be determined as follows:

$$\Re_B = \sum_{n=1}^{N_B} (n\, R_B)\pi_B(n) = \lambda_B C_B(1 - \Im_B).$$

Maximization of revenue

It is desirable to maximize the mean revenue with respect to the design parameters δ and N_B while fulfilling the given availability specification. We formulate this problem as follows:

$$\text{maximize } \Re = \Re_I + \Re_B \tag{16a}$$

$$\text{subject to } 1.\ \Im_I \leqslant \Im_I^* \tag{16b}$$

$$2.\ \Im_B \leqslant \Im_B^* \tag{16c}$$

$$3.\ N_I + N_B = N \tag{16d}$$

$$4.\ 0 \leqslant N_B \leqslant N \tag{16e}$$

$$5.\ 0 < \delta \leqslant 1. \tag{16f}$$

The objective function is the mean total revenue $\Re$. The first and second constraints ensure the given availability specification is met. The third and fourth constraints ensure that N_I and N_B sum to N and are non-negative, respectively. The fifth constraint ensures that δ is positive and is not larger than one. The optimal δ, N_B and $\Re$ are denoted by δ^*, N_B^* and $\Re^*$, respectively.

For a rational charging scheme, the number of possible values of δ should be small. For example, δ can assume such values as 0.95, 0.90 or 0.85 (ie 5%, 10% or 15% discount), but it should not assume any arbitrary values such as 0.917 or 0.873. Therefore, we can enumerate all possible values of δ for optimization, so that the resulting algorithm is applicable to any relationship between β and δ. In each enumeration, we determine the feasible values of N_B to fulfil equations (16*b*)–(16*e*), and then maximize $\Re$ with respect to N_B.

We determine the feasible values of N_B to fulfill equations (16*b*)–(16*e*) as follows. It is clear that $\Im_B$ is a decreasing function of N_B and hence N_B must be larger than a certain value to fulfil the availability specification $\Im_B \leqslant \Im_B^*$. In other words, this specification can be regarded as a lower bound *LB* on N_B such that $N_B \geqslant LB$. To compute *LB*, we start with the interval $[0, N]$. If $\Im_B|_{N_B=\lfloor \frac{N}{2} \rfloor} \leqslant \Im_B^*$, then *LB* must be in the interval $\left[0, \lfloor \frac{N}{2} \rfloor\right]$; otherwise, *LB* must be in the interval $\left[\lfloor \frac{N}{2} \rfloor, N\right]$. In this iteration, we reduce the width of the interval by about 50%. We repeat this step iteratively until the width is reduced to one. The details are given in procedure `compute_LB` in Appendix A.

Since $\Im_I$ is a decreasing function of N_I and $N_B = N - N_I$, $\Im_I$ is an increasing function of N_B. The availability specification $\Im_I \leqslant \Im_I^*$ can be regarded as an upper bound *UB* on N_B such that $N_B \leqslant UB$. We compute *UB* in a similar manner and the details are given in procedure `compute_UB` in Appendix B.

After computing *LB* and *UB*, the feasible range of N_B to fulfil equations (16*b*)–(16*e*) is $LB \leqslant N_B \leqslant UB$. We now maximize $\Re$ with respect to N_B within the range $LB \leqslant N_B \leqslant UB$, and we need the following lemma.

Lemma 1 Let n be any positive integer, ρ be any positive real number, and

$$f(n, \rho) \equiv \frac{\sum_{j=0}^{n-1} \frac{\rho^j}{j!}}{\sum_{j=0}^{n} \frac{\rho^j}{j!}}$$

$f(n, \rho)$ is a concave function of n.

Proof The proof can be found in ref. 20.

We express $\Re_I$ and $\Re_B$ in terms of $f(\bullet, \bullet)$:

$$\begin{cases} \Re_I = \left(\lambda(1-\beta) \sum_{i=1}^{M} C_i\alpha_i\right) f(N_I, \lambda_I\, \overline{\tau}_I) \\ \Re_B = \left(\delta\lambda\beta \sum_{i=1}^{M} C_i\alpha_i\right) f(N_B, \lambda_B\, \overline{\tau}_B). \end{cases} \tag{17}$$

Since $\delta, \lambda, \beta, C_i$ and α_i are independent of $N_B, \Re_B$ is a concave function of N_B and $\Re_I$ is a concave function of N_I. Since $N_B = N - N_I$ where N is given and fixed, $\Re_I$ is also a concave function of N_B. Therefore, the mean total revenue $\Re = \Re_I + \Re_B$ is a concave function of N_B. To maximize $\Re$ with respect to N_B within the interval $LB \leqslant N_B \leqslant UB$, we reduce the width of the interval iteratively. Specifically, we use the golden ratios 0.382 and 0.618[21] to divide the interval into golden intervals, and then identify the one containing the optimal N_B. We repeat this step iteratively until we cannot further divide the interval. The details are given in procedure `optimize_NB` in Appendix C.

The overall steps for maximizing the mean total revenue subject to the given availability specification are given as follows:

[Inputs: $N, \Im_I^*, \Im_B^*, \beta, C_i, \alpha_i, \tau_i, \overline{\tau}_i$ for $1 \leqslant i \leqslant M$]
[Outputs: $\Re^*$, δ^* and N_B^*]

1. $\Im^* = 0$.
2. **FOR** each possible value of δ **DO**
   BEGIN

2.1 execute `compute_LB` to find the lower bound *LB* on N_B;
2.2 execute `compute_UB` to find the upper bound *UB* on N_B;
2.3 **IF** there are feasible *LB* and *UB* **THEN**
BEGIN

2.3.1 execute `optimize_NB` to maximize $\Re$ with respect to N_B within the range $LB \leqslant N_B \leqslant UB$;
2.3.2 IF $\Re > \Re^*$ **THEN** $\Re^* = \Re$, $N_B^* = N_B$ and $\delta^* = \delta$
END
END

Remark 4. The optimal relative service charge δ depends on the arrival rate λ. At different time of the day, λ may be different (eg λ is larger in the peak hours of every night), and we can choose a different optimal δ for each period of time.

Remark 5. As the VOD service becomes more popular, the arrival rate becomes larger. We can cope with this larger arrival rate as follows. If the availability specifications can still be fulfilled, it is only necessary to optimize δ. Otherwise, it is necessary to increase the capacity N of the system as well as optimize δ.

Numerical results

There are $M = 100$ video programmes, and their popularity follows the Zipf distribution:[22] $\alpha_i = A/i^{0.729}$ where A is a normalization constant and the parameter value 0.729 was found by empirical measurement.[9,10,23] We let $C_1 = C_2 = \cdots = C_{20} = 30$, $C_{21} = C_{22} = \cdots = C_{100} = 20$, $\tau_i = 90$ min and $\overline{\tau}_i = 100$ min for $1 \leqslant i \leqslant 100$, $W = 10$ min, $\Im_I^* = 0.02$ and $\Im_B^* = 0.01$. The possible values of δ and the relationship between β and δ are given in Table 1. For comparison, we also compute the mean revenue and the blocking probability when there is no incentive charging.

Figure 4 shows the effectiveness of incentive charging for different λ when $N = 1000$. The curves for incentive charging are not smooth because δ assumes discrete values. In general, compared with the case without incentive charging, the incentive charging scheme generates more revenue for the service provider (Figure 4(a)), provides a lower blocking probability to the users (Figure 4(b)), and imposes a lower service charge on the batch service (Figure 4(c)). For example, when $\lambda = 20$ requests/min, the scheme increases the revenue by 48%, reduces the blocking probability by about six orders of magnitude, and reduces the service charge of batch service by 35%. In particular, the scheme is more effective when λ is large. For example, as λ increases from 10 to 30 requests/min, the mean revenue $\Re$ increases from \$HK263.3/min to \$HK539.7/min and the relative service charge decreases from 0.90 to 0.60. Therefore, the incentive charging scheme is particularly effective in peak hours (say, 7:00–10:00 pm) when the demand for VOD service is large.

Figure 5 shows the effectiveness of incentive charging for different N when $\lambda = 20$ requests/min. To generate the same revenue, Figure 5(a) shows that the scheme requires a smaller number of video streams (ie a smaller system cost). At the same time, it still gives a smaller blocking probability (Figure 5(b)) and provides a smaller service charge for the batch service (Figure 5(c)). For example, to generate a revenue of \$HK300/min, the scheme reduces the required number of video streams from 1120 to 560. From this viewpoint, the incentive charging scheme can effectively reduce the cost of the VOD system while generating the same revenue.

Conclusions

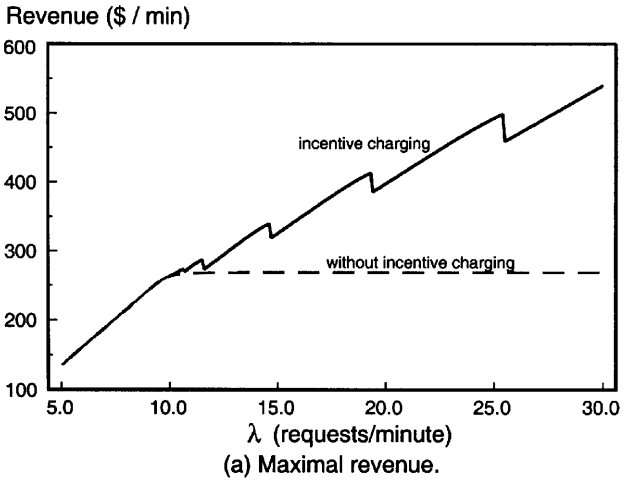
In this paper, we considered a VOD system providing an individual service offering interactive facilities, as well as a batch service offering a lower service charge, and proposed an incentive charging scheme to optimize their coexistence. By imposing a lower service charge on the batch service, the incentive charging scheme is advantageous to both the service provider and the users:

- The service provider can get more revenue by serving more concurrent users via the batch service. Equivalently, the system requires a smaller number of video streams to generate the same revenue (ie the system cost is smaller). In addition, the service provider can provide a higher availability to the users because the incentive charging scheme can reduce the blocking probability.
- The users can choose their preferred services. If they do not need interactive operations and they accept a small start-up delay, they can enjoy a lower service charge. They can also experience a higher availability.

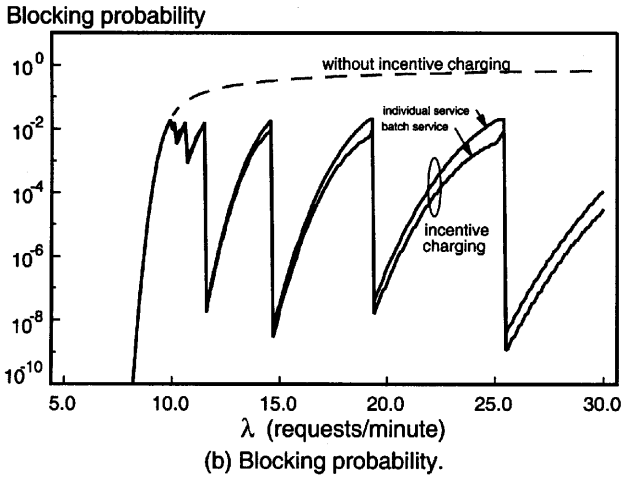
We analyzed the incentive charging scheme and maximized the mean revenue subject to the given availability specification. The numerical results show that the scheme is particularly effective in peak hours when the demand for VOD service is large.

Table 1 Relationship between β and δ in the numerical experiment

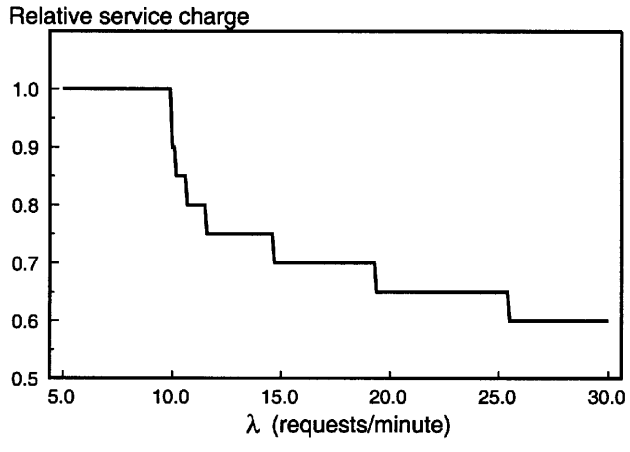
δ	β
0.50	0.95
0.55	0.90
0.60	0.83
0.65	0.75
0.70	0.65
0.75	0.50
0.80	0.30
0.85	0.20
0.90	0.12
0.95	0.05
1.00	0.00



(a) Maximal revenue.

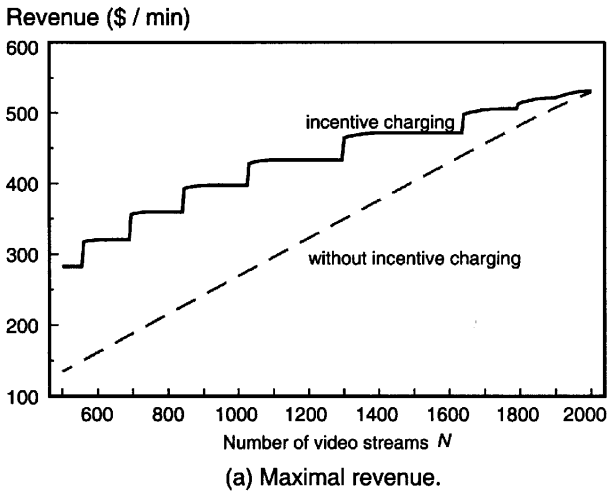


(b) Blocking probability.

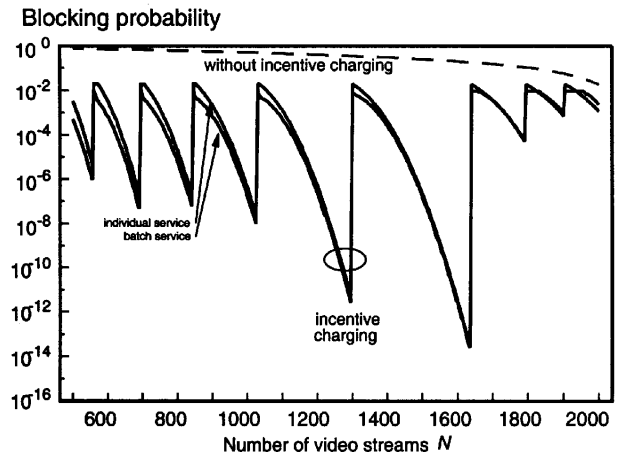


(c) Optimal relative service charge.

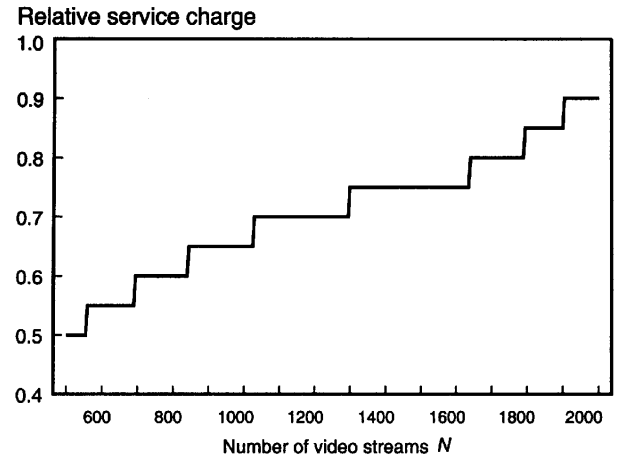
Figure 4 Optimal incentive charging under different demand ($N = 1000$). (a) Maximal revenue. (b) Blocking probability. (c) Optimal relative service charge.



(a) Maximal revenue.



(b) Blocking probability.



(c) Optimal relative service charge.

Figure 5 Optimal incentive charging for different number of available video streams ($\lambda = 20$ requests/min). (a) Maximal revenue. (b) Blocking probability. (c) Optimal relative service charge.

Appendix A. Procedure `compute_LB`

$$\begin{bmatrix} \text{Inputs:} & \Im_{\mathrm{B}}^{*}, N, \lambda_{\mathrm{B}}, \overline{\tau}_{\mathrm{B}} \\ \text{Output:} & LB \end{bmatrix}$$

1. **IF** $\Im_{\mathrm{B}} |_{N_{\mathrm{B}}=N} > \Im_{\mathrm{B}}^{*}$ **THEN** there is no feasible LB and stop
2. **IF** $\Im_{\mathrm{B}} |_{N_{\mathrm{B}}=0} \leqslant \Im_{\mathrm{B}}^{*}$ **THEN** $LB = 0$ and stop
3. $x_l = 0$ and $x_u = N$
4. **WHILE** $(x_u - x_l) > 1$ **DO**
   BEGIN

   $$x = \left\lfloor \frac{x_u + x_l}{2} \right\rfloor;$$

   IF $\Im_{\mathrm{B}} |_{N_{\mathrm{B}}=x} \leqslant \Im_{\mathrm{B}}^{*}$ **THEN** $x_u = x$ **ELSE** $x_l = x$;
   END
5. $LB = x_u$

Appendix B: Procedure `compute_UB`

$$\begin{bmatrix} \text{Inputs:} & \Im_{\mathrm{I}}^{*}, N, \lambda_{\mathrm{I}}, \overline{\tau}_{\mathrm{I}} \\ \text{Output:} & UB \end{bmatrix}$$

1. **IF** $\Im_{\mathrm{I}} |_{N_{\mathrm{B}}=0} > \Im_{\mathrm{I}}^{*}$ **THEN** there is no feasible UB and stop
2. **IF** $\Im_{\mathrm{I}} |_{N_{\mathrm{B}}=N} \leqslant \Im_{\mathrm{I}}^{*}$ **THEN** $UB = N$ and stop
3. $x_l = 0$ and $x_u = N$
4. **WHILE** $(x_u - x_l) > 1$ **DO**
   BEGIN

   $$x = \left\lfloor \frac{x_u + x_l}{2} \right\rfloor;$$

   IF $\Im_{\mathrm{I}} |_{N_{\mathrm{B}}=x} \leqslant \Im_{\mathrm{I}}^{*}$ **THEN** $x_l = x$ **ELSE** $x_u = x$;
   END
5. $UB = x_l$

Appendix C. Procedure `optimize_NB`

$$\begin{bmatrix} \text{Inputs:} & LB, UB, \beta, \delta, C_i, \alpha_i, \tau_i, \overline{\tau}_i \text{ for } 1 \leqslant i \leqslant M \\ \text{Outputs:} & \Re, N_{\mathrm{B}} \end{bmatrix}$$

1. $a = LB, c = UB$, and use the bisection method to select b such that $a < b < c$, $\Re |_{N_{\mathrm{B}}=a} < \Re |_{N_{\mathrm{B}}=b}$ and $\Re |_{N_{\mathrm{B}}=b} > \Re |_{N_{\mathrm{B}}=c}$. If there does not exist b fulfilling these conditions, choose N_{B} to be

   $$N_{\mathrm{B}} = \begin{cases} a & \text{if } \Re |_{N_{\mathrm{B}}=a} \geqslant \Re |_{N_{\mathrm{B}}=c} \\ c & \text{if } \Re |_{N_{\mathrm{B}}=c} > \Re |_{N_{\mathrm{B}}=a} \end{cases}$$

   and stop
2. **IF** $| c - b | > | b - a |$
   THEN
   $N_0 = a, N_1 = b, N_2 = \lfloor 0.618b + 0.382c \rfloor$, and $N_3 = c$
   ELSE
   $N_0 = a, N_1 = \lfloor 0.382a + 0.618b \rfloor, N_2 = b$ and $N_3 = c$
3. **WHILE** $N_3 - N_0 > 3$ **DO**
   BEGIN
   IF $\Re |_{N_{\mathrm{B}}=N_2} > \Re |_{N_{\mathrm{B}}=N_1}$
   THEN $N_0 = N_1$, $N_1 = N_2$,
   $N_2 = \lfloor 0.618N_1 + 0.382N_3 \rfloor$
   ELSE $N_3 = N_2$, $N_2 = N_1$,
   $N_1 = \lfloor 0.382N_0 + 0.618N_2 \rfloor$
   END
4. **IF** $\Re |_{N_{\mathrm{B}}=N_1} > \Re |_{N_{\mathrm{B}}=N_2}$ **THEN** $N_{\mathrm{B}} = N_1$ and $\Re = \Re |_{N_{\mathrm{B}}=N_1}$
   ELSE $N_{\mathrm{B}} = N_2$ and $\Re = \Re |_{N_{\mathrm{B}}=N_2}$

Acknowledgements—We sincerely thank the editor, Dr John Ranyard and the anonymous reviewers for their insightful comments and helpful suggestions.

References

1 Deloddere D, Verbiest W and Verhille H (1994). Interactive video on demand. *IEEE Commun Mag* **32**: 82–88.
2 Li VOK and Liao WJ (1997). Distributed multimedia systems. *Proc IEEE* **85**: 1063–1108.
3 http://www.adsl.com/vdsl_tutorial.html
4 http://www.adsl.com/adsl_tutorial.html
5 *Hong Kong Sing Tao Daily Newspaper* (in Chinese), Hong Kong, 29 April 1997.
6 http://www.netvigator.com/IMS
7 http://www.itvhk.com
8 Anderson DP (1993). Metascheduling for continuous media. *ACM Trans Comput Syst* **11**: 226–252.
9 Dan A, Sitaram D and Shahabuddin P (1994). Scheduling policies for an on-demand video server with batching. In: Proceedings of the ACM Multimedia Conference and Exposition, ACM Press, USA, pp 15–23.
10 Dan A, Shahabuddin P, Sitaram D and Towsley D (1995). Channel allocation under batching and VCR control in video-on-demand systems. *J Parallel Dist Comput* **30**: 168–179.
11 Almeroth KC and Ammar MH (1996). The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE J Sel Areas Commun* **14**: 1110–1122.
12 Li VOK, Liao WJ, Qiu X and Wong EWM (1996). Performance model of interactive video-on-demand systems. *IEEE J Sel Areas Commun* **14**: 1099–1109.
13 Liao WJ and Li VOK (1997). The split-and-merge (SAM) protocol for interactive video-on-demand systems. In: Proceedings of the IEEE INFOCOM, Japan.
14 Rao S and Petersen ER (1998). Optimal pricing of priority services. *Oper Res* **46**: 46–56.
15 Leung YW and Wong EWM (1998). An incentive charging scheme for video-on-demand. *Technical Report*, Hong Kong Baptist University.
16 *Hong Kong Economic Journal*, 15 August 1999.
17 Ross SM (1983). *Stochastic Processes.* Wiley: Chichester.
18 Gross D and Harris CM (1985). *Fundamentals of Queueing Theory.* Wiley: Chichester, pp 294–304.
19 Kaufman JS (1981). Blocking in a shared resource environment. *IEEE Trans Commun* **29**: 1474–1481.
20 Jagers AA and Doorn EAV (1986). On the continued Erlang loss function. *Oper Res Lett* **5**: 43–46.
21 Press WH, Flannery BP, Teukolsky SA and Vetterling WT (1992). *Numerical Recipes in Pascal: The Art of Scientific Computing.* Cambridge University Press: Cambridge.
22 Zipf GK (1949). *Human Behaviour and the Principle of Least Effort.* Addison-Wesley: New York.
23 *Video Store Magazine*, Dec. 13, 1992.

Received May 1999;
accepted August 2000 after three revisions