

Multicast Distribution for Multimedia on Demand Services

G. Boggia, P. Camarda*, P. L. Mazzeo, M. Mongiello

Politecnico di Bari – Dip. di Elettrotecnica ed Elettronica
Via. E. Orabona n.4 – 70125 Bari (Italy)
Tel. +39 080 5460642, fax +39 080 5460410
e-mail: camarda@poliba.it

Abstract

Recent advances in Information and Communication technologies have made multimedia on demand services technically and economically feasible. An important aspect of such systems is the resource sharing technique, which allow the contemporaneous service of a large number of user requests with a considerable saving in terms of network bandwidth and server resources. In this paper, we report the results of a study which analyzes batching and buffering techniques to group and serve together video requests. The mathematical model allows the evaluation of the main system performance (probability distribution of the number of streams, percentage reduction of resources, etc.) as a function of load and batching interval duration. Simulation experiments confirm the analytical model in the whole range of considered conditions.

Key words : Multimedia Services, Batching and Buffering Techniques, Analytical Models

* Corresponding author

Multicast Distribution for Multimedia on Demand Services

G. Boggia, P. Camarda, P. L. Mazzeo, M. Mongiello

Politecnico di Bari – Dip. di Elettrotecnica ed Elettronica
Via. E. Orabona n.4 – 70125 Bari (Italy)
Tel. +39 080 5460642, fax +39 080 5460410
e-mail: camarda@poliba.it

Abstract

Recent advances in Information and Communication technologies have made multimedia on demand services technically and economically feasible. An important aspect of such systems is the resource sharing technique, which allow the simultaneous service of a large number of user requests with a considerable saving in terms of network bandwidth and server resources. In this paper, we report the results of a study which analyzes batching and buffering techniques, that is grouping and serving together video requests. The mathematical model allows the evaluation of the main system performance (probability distribution of the number of streams, percentage reduction of resources, etc.) as a function of load and batching interval duration. Simulation experiments confirm the analytical model in the whole range of considered conditions.

I Introduction

Multimedia systems providing on demand services, as Video on Demand (VoD), distance learning, internet video broadcast, etc. are now full feasible thanks to recent advances in networking and computer technologies. One of the most challenging aspects of such systems is the architecture design for providing the required service with the appropriate Quality of Service (QoS). QoS is determined by several parameters (such as I/O bandwidth, memory requirements, etc.) which behave as critical components in designing system architecture; in fact, the system must satisfy the real-time constraints for continuous delivery of video objects at a specified bandwidth.

There are many approaches to improve the efficiency of Multimedia on Demand systems by managing critical resources [1]. These include disk scheduling and data placement algorithms, techniques for memory management or strategies for CPU and other resources scheduling, data and resources sharing [2]. Several approaches for optimizing system performance, by sharing the available resources, consider, as critical resource, the I/O bandwidth required to satisfy the client request. The objective is to reduce the bandwidth demand increasing the number of client requests which

can be served simultaneously. The most important techniques proposed to implement the mentioned optimization comprehend batching of requests, merging of video streams, buffering of central memory and patching schemes.

In the batching technique [1] [3], client requests for the same video, emitted during a short interval of time, i.e. the batching interval, are grouped in a *batch* and the display of the video is delayed for the batching interval. Thus, multiple customers requiring the same video within this batching interval can be served by multicasting the same video stream, with considerable savings in server and network capacity.

In merging strategies, the video streams are merged into a single one by adjusting the display rates of requests for the same object. Some applications of these techniques are described in [4] [5].

The third method, i.e., the buffering technique, consists in using the central memory as buffers, holding a given number of frames behind a video stream. Subsequent requests for that video, within the batching interval, can be served by using the buffer rather than requesting another I/O stream to the server [6] [8].

The patching scheme relies on more sophisticated user apparatuses which must be able to receive two streams simultaneously. The user request is served by an existing stream for that video object (if there is any) which is buffered in the user apparatus and simultaneously a new stream is requested to the server for the frames already transmitted [9] [10].

These functions may be implemented in a multitiered hierarchical distributed video server as depicted in Fig.1. Through a Residential Access Network (RAN), clients are connected to a Local Switching Office (LSO) storing the most popular video programs. User requests can be satisfied by the LSO or may be forwarded either to Regional Video Servers through a Metropolitan Area Network (MAN), or to remote servers through a backbone network [1]. Alternatively, the previous functions can be implemented in a standard internet environment [11].

The techniques mentioned before can be exploited at various levels of the hierarchy. In this paper we analyze system performance in the hypotheses of batching and buffering techniques applied at one level of the hierarchy. In particular, we evaluate the probability distribution of the number of batches and the percentage reduction of resources as a function of load and batching interval.

The paper is organized as follow. Section II provides a customer behavior model based on a queuing network, introducing the performance evaluation parameters used to investigate the model performance. In Section III we study the effectiveness of the model and achieve numerical results; besides we simulate the system operations and discuss simulation results as a measure to validate the proposed model. Section IV presents the concluding remarks.

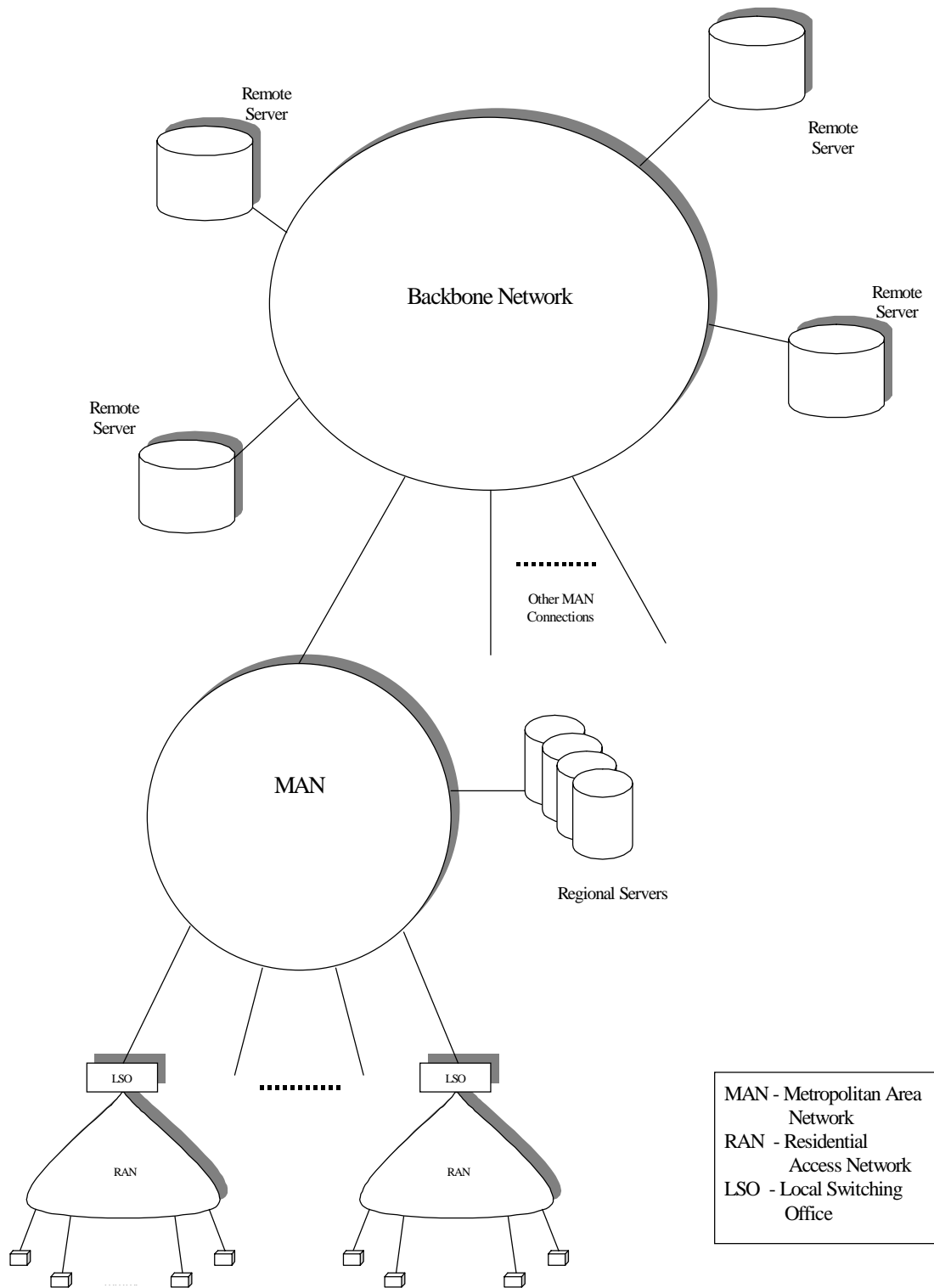


Figure 1. Multimedia on Demand System

II System Modeling

II.1 System Description

In this section we describe an analytical model to study users behavior in a multimedia system which adopts buffering or batching techniques.

We consider N_U users connected to the multimedia system. Each free user generates a new video request following an exponential distribution with parameter λ [requests/min]. Let π_c and S_c be, respectively, the request probability and the duration of video c , with $c=1, 2, \dots, N_F$, where N_F is the total number of available video programs in the system. For sake of simplicity, we suppose that the only interactive user operation is the video request, i.e., there are no VCR-like operations (e.g., pause, stop, fast-forward, rewind, etc.). Let N_{AU} be the maximum number of simultaneous active users, i.e., users with a video display in progress.

The analysis is conducted following two phases. In the first one the number of requests for each video program is evaluated using standard queuing models. The second phase, based on the results of the first one evaluates the main system performance parameters (probability distribution of the number of batches, percentage reduction of resources, etc.) as a function of load, blocking interval, etc. The analytical results are validated by simulation in the considered context.

As regard the first phase, we describe the system with a closed queueing network (see Fig. 2) composed by two service centers and N_F+1 classes. The first center considers users belonging to class 0, which corresponds to the *idle state*. It can be modeled by an Infinite Servers (IS) service center, with exponential service time. The second center models users in the *active state*. Each user belongs to a class which corresponds to the requested video. At service completion, the user returns in the first center at class 0. Also this second service center can be modeled as an Infinite Server (IS) service center, with a constraint on the global number of users which cannot exceed the maximum value N_{AU} . The service time in this second center is supposed general with an average value T_c . The hypothesis of infinite servers means that there are no upper limits in the number of permitted streams. This unusual choice is determined by the consideration that this model has been developed as an aid to the synthesis (rather than in the analysis) of a multimedia system.

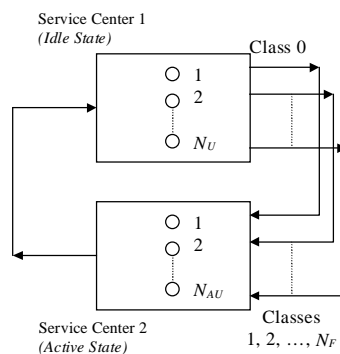


Figure 2. System modeling with queueing network

After the first request, all users that make requests for the same video in the batching interval t_B are grouped together sharing the system resources. With the buffering technique, the allocated resources for each group are released when the last user ends video display. Then the service time of

an active class c user is equal to the deterministic video duration S_c . With the batching technique, users belonging to the same batch are served together and video display starts at the end of the batching interval. In this case, the service time is general with an average value given by the sum of the deterministic video duration and the mean waiting time before video start. Since the incoming video c request represents a random point within the batching interval, the mean waiting time is $t_B/2$ [12], then the mean service time is $S_c+t_B/2$.

II.2 Distribution of idle and active users

By hypothesis, the crossing class probabilities are:

$$p_{0,0} = 0; \quad p_{0,c} = \pi_c; \quad p_{k,0} = 1; \quad p_{k,c} = 0; \quad (1)$$

with $c = 1, \dots, N_F$ and $k = 1, \dots, N_F$.

Let $n_{(1)}$ and $n_{(2)}$ be, respectively, the number of users at the first center and at the second one; let n_c , with $c = 0, 1, \dots, N_F$, be the number of users at class c . To solve the queueing network, the first step is the resolution of the following homogeneous system of linear equations [13], which admits infinite solutions:

$$y_c = \sum_{d=0}^{N_F} y_d p_{d,c} \quad c = 0, 1, \dots, N_F \quad (2)$$

thus, the relative service time at class c is:

$$b_c = y_c \cdot T_c \quad c = 0, 1, \dots, N_F \quad (3)$$

and the relative service times at service centers are:

$$b_{(1)} = y_0 \cdot T_0 \quad b_{(2)} = \sum_{c=1}^{N_F} y_c T_c \quad (4)$$

In the stated hypothesis, the network admits a product form solution [14] and the state probability is given by:

$$P(n_{(1)}, n_{(2)}) = \frac{1}{G} \frac{b_{(1)}^{n_{(1)}} b_{(2)}^{n_{(2)}}}{n_{(1)}! n_{(2)}!} \quad \text{with } (n_{(1)}, n_{(2)}) \in \mathcal{S} \quad (5)$$

where $\mathcal{S} = \{(n_{(1)}, n_{(2)}) \mid n_{(1)} + n_{(2)} = N_U, n_{(2)} \leq N_{AU}\}$ is the states space and G is the normalization constant. Since $n_{(1)} = N_U - n_{(2)}$ and the maximum value for $n_{(2)}$ is N_{AU} , we have [14]:

$$G = \sum_{(n_{(1)}, n_{(2)}) \in \mathcal{S}} \frac{b_{(1)}^{n_{(1)}} b_{(2)}^{n_{(2)}}}{n_{(1)}! n_{(2)}!} = \sum_{i=0}^{N_{AU}} \frac{b_{(1)}^{N_U-i} b_{(2)}^i}{(N_U - i)! i!} \quad (6)$$

Once this constant has been found, it allows the evaluation of the marginal state probability for the number of users at each center:

$$P(n_{(1)}=i) = P(n_{(1)}=i, n_{(2)} = N_U - i) \quad \text{with } i = N_U - N_{AU}, \dots, N_U$$

$$P(n_{(2)}=i) = P(n_{(1)}=N_U - i, n_{(2)} = i) \quad \text{with } i = 0, \dots, N_{AU} \quad (7)$$

and the marginal state probability for the number of users at each class c [13] with $c = 1, \dots, N_F$:

$$P(n_c = i) = \sum_{k=i}^{N_{AU}} \frac{k!}{i!(k-i)!} \frac{b_c^i (b_{(2)} - b_c)^{k-i}}{b_{(2)}^k} P(n_{(2)} = k). \quad (8)$$

II.3 Batch Modeling

The second phase is based on the results of the first one and allows the evaluation of the probability distribution of the number of batches and the related performance indices.

Let n_{Bc} be the number of batches for the video c , with $c = 1, \dots, N_F$. By the total probability theorem, we have:

$$P(n_{Bc} = i) = \sum_{j=0}^{N_{AU}} P(n_{Bc} = i | n_c = j) P(n_c = j) \quad (9)$$

with $c = 1, \dots, N_F$ and $i=0, \dots, N_{AU}$.

It is to note that:

- $P(n_{Bc} = 0/n_c=j) = 0 \quad j \neq 0$; with active users, the minimum value for n_{Bc} is 1, i.e., all users require the same video in the batching interval;
- $P(n_{Bc} = 0/n_c=0) = 1$ without active users, certainly, n_{Bc} is 0;
- $P(n_{Bc} = i/n_c=0) = 0 \quad i \neq 0$; without active users, there are no batches;
- $P(n_{Bc} = i/n_c=j) = 0 \quad i > j$; maximum value for n_{Bc} is j ; in this case we have batches with single user, that is, really, there are no groups of users, i.e., no real batches.

We need the conditional probability $P(n_{Bc} = i | n_c = j)$ for $i=1, \dots, N_{AU}$ and $j = i, \dots, N_{AU}$. Let us consider i batches with j active users of class c ; let $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_i)$ be the vector of the number of users for each batch of class c , where α_k is the number of users of the k^{th} batch and $\alpha_1 + \alpha_2 + \dots + \alpha_i = j$. Summing over all possible state α , we obtain:

$$P(n_{Bc} = i | n_c = j) = \sum_{\alpha_1 + \alpha_2 + \dots + \alpha_i = j} P(\alpha_1, \alpha_2, \dots, \alpha_i). \quad (10)$$

Now, referring to the time diagram for request arrivals in Fig. 3, where t_k is the instant of the k^{th} user arrival, the probability $P(\alpha)$ is given by:

$$P(\alpha_1, \alpha_2, \dots, \alpha_i) = P(t_{\alpha_1} - t_1 \leq t_B, t_{\alpha_1+1} - t_1 > t_B) \cdot P(t_{\alpha_1+\alpha_2} - t_{\alpha_1+1} \leq t_B, t_{\alpha_1+\alpha_2+1} - t_{\alpha_1+1} > t_B) \cdot \dots$$

$$\cdot P(t_{\alpha_1+\dots+\alpha_{i-1}} - t_{\alpha_1+\dots+\alpha_{i-2}+1} \leq t_B, t_{\alpha_1+\dots+\alpha_{i-1}+1} - t_{\alpha_1+\dots+\alpha_{i-2}+1} > t_B) \cdot P(t_{\alpha_1+\dots+\alpha_i} - t_{\alpha_1+\dots+\alpha_{i-1}+1} \leq t_B) . \quad (11)$$

Each of the first $i-1$ terms is the probability that a time period t_q-t_p , with $1 \leq p \leq q \leq j$, is less or equal than t_B with the time period $t_{q+1}-t_p$ greater than t_B . This probability (see Appendix A) is given by:

$$\Phi(q-p+1) = P(t_q-t_p \leq t_B, t_{q+1}-t_p > t_B) = \frac{e^{-\lambda_c \cdot t_B} (\lambda_c t_B)^{q-p}}{(q-p)!} \quad (12)$$

where λ_c is the accepted requests rate for video c at the active state center, that will be evaluated later.

The last term of (11) is the probability that the time period between a first arrival and an x^{th} one, $t_{p+x} - t_{p+1}$, is less or equal than t_B ; then, simply by definition, this time period is an Erlang random variable [12] with parameters $x-1$ and λ_c , thus:

$$\Psi(x) = P(t_{p+x}-t_{p+1} \leq t_B) = 1 - \sum_{k=0}^{x-2} \frac{e^{-\lambda_c \cdot t_B} (\lambda_c t_B)^k}{k!} . \quad (13)$$

Using (12) and (13), (11) becomes:

$$P(\alpha) = \frac{e^{-\lambda_c \cdot t_B} (\lambda_c t_B)^{\alpha_1-1}}{(\alpha_1-1)!} \dots \frac{e^{-\lambda_c \cdot t_B} (\lambda_c t_B)^{\alpha_{i-1}-1}}{(\alpha_{i-1}-1)!} \left(1 - \sum_{k=0}^{\alpha_i-2} \frac{e^{-\lambda_c \cdot t_B} (\lambda_c t_B)^k}{k!} \right) = \Phi(\alpha_1) \dots \Phi(\alpha_{i-1}) \Psi(\alpha_i) . \quad (14)$$

Thus, we have:

$$P(n_{Bc} = i / n_c = j) = \sum_{\alpha_1+\alpha_2+\dots+\alpha_i=j} P(\alpha_1, \alpha_2, \dots, \alpha_i) = \sum_{\alpha_1+\alpha_2+\dots+\alpha_i=j} \Phi(\alpha_1) \dots \Phi(\alpha_{i-1}) \Psi(\alpha_i) . \quad (15)$$

Practically, we can evaluate this expression, with low computational costs, using a simple iterative algorithm (see Appendix B).

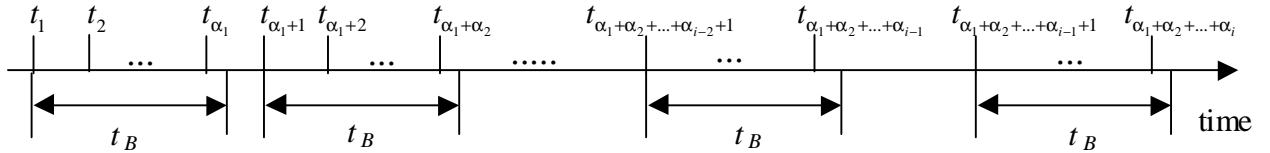


Figure 3. Time diagram for request arrivals

To evaluate the rate λ_c , we suppose that each idle user generates a new request, and thus transits in the active state, following an exponential distribution with parameter λ ; the average accepted request rate at the active state center is:

$$\lambda_F = \lambda \cdot E[n_R] \quad (16)$$

where $E[n_R]$ is the mean number of accepted video requests. Since no other requests are accepted

when the active center is full, the random variable n_R assumes the values:

$$n_R = \begin{cases} (N_U - n_{(2)}) & n_{(2)} < N_{AU} \\ 0 & n_{(2)} = N_{AU} \end{cases}. \quad (17)$$

Its average value is:

$$E[n_R] = \sum_{i=0}^{N_{AU}-1} (N_U - i) \cdot P(n_{(2)} = i) = \sum_{i=0}^{N_{AU}} (N_U - i) \cdot P(n_{(2)} = i) - (N_U - N_{AU})P(n_{(2)} = N_{AU}) \quad (18)$$

then:

$$E[n_R] = N_U - E[n_{(2)}] - (N_U - N_{AU})P(n_{(2)} = N_{AU}) \quad (19)$$

where $E[n_{(2)}]$ is the mean number of active users.

By the splitting property of Poisson process [13], for class c , the accepted request rate at the active center is:

$$\lambda_c = \lambda_F \cdot \pi_c \quad (20)$$

where π_c is the video c request probability.

II.4 Performance Indices

At this point we define some parameters for evaluating the performance of the analyzed multimedia system.

The first performance parameter is the probability of unsuccessful video request, P_U , i.e., the probability that a user, making a video request, does not find available system resources because there are already N_{AU} active users. Taking into account the arrival theorem [15], when a user makes a service request, in the transition instant from idle to active center, we must consider the system with N_U-1 users instead of N_U . Thus, the unsuccessful probability is:

$$P_U = P(n_{(2)} = N_{AU}) \quad \text{with } N_U-1 \text{ users in the system.} \quad (21)$$

Another parameter is the percentage reduction of resources requirement, $R\%$:

$$R\% = \left(1 - \frac{E[n_B]}{E[n_{(2)}]} \right) \cdot 100 \quad (22)$$

where $E[n_B]$ is the mean number of batches in the system:

$$E[n_B] = \sum_{c=1}^{N_F} E[n_{Bc}] = \sum_{c=1}^{N_F} \sum_{i=0}^{N_{AU}} i \cdot P(n_{Bc} = i) \quad (23)$$

and $E[n_{(2)}]$ is the mean number of active users given by:

$$E[n_{(2)}] = \sum_{i=0}^{N_{AU}} i \cdot P(n_{(2)} = i). \quad (24)$$

III Numerical results

In this section we provide numerical results of the analytical model and compare them with the results obtained from simulation in a simple VoD system. We consider $N_U = 100$ users, $N_F = 10$ available video programs, a given request probability and a video duration per each video (see Table I). The results illustrated are referred to the buffering techniques, the application to batching may be easily obtained as described in section II by considering as the mean service time the value $S_c + t_B/2$.

The behavior of the mean number of batches, $E[n_B]$, is observed as a function of the video request rate per user, λ [requests/min], the batching interval, t_B [min] and the maximum number of active users, N_{AU} . Other important results are concerned with $R\%$, that is the percentage reduction of resources requirement when resources are shared among the users of a batch, with respect to the direct allocation of resource to users; $R\%$ behavior is shown as a function of λ with parameter t_B and N_{AU} . Moreover, we investigate the unsuccessful video request probability P_U as a function of the arrival rate λ and for different value of N_{AU} and as a function of λ with N_{AU} as a parameter.

In all the following figures, the plain lines represent analytical results, while the simulation results are reported using the symbol '+'.

TABLE I

Video	1	2	3	4	5	6	7	8	9	10
Request Probability (π_i)	0.2	0.2	0.15	0.15	0.075	0.075	0.075	0.025	0.025	0.025
Video Duration S_i [min]	120	90	85	100	60	120	90	75	90	120

III.1 Mean number of active users and batches: $E[n_{(2)}]$ and $E[n_B]$

A comparison of the mean number of active users, $E[n_{(2)}]$, and $E[n_B]$ is illustrated in Fig. 4, where the maximum number of active users is $N_{AU} = 60$.

We note that $E[n_B]$ approaches to $E[n_{(2)}]$ for λ close to zero. This means that, for small values of λ , the probability of more than one request for the same video in the time interval t_B is not relevant, then each batch has approximately one user. Note that the differences between the two curves become larger when λ grows, here $E[n_B]$ assumes increasing values but keeps always below $E[n_{(2)}]$. This is due to the fact that there is a higher number of active users that are grouped in batches.

Figures 5 and 6 represent respectively the dependence of $E[n_{(2)}]$ and of $E[n_B]$ on the maximum number of active users. In these figures, we consider a batching interval of $t_B = 10$ minutes. The difference between the two curves becomes meaningful for large values of λ where the system has

high blocking probability. For decreasing value of N_{AU} each curve reaches the saturation for smaller values of λ . On the other hand, for a fixed λ a reduction in the number of active users causes a reduction in the number of batches then for smaller N_{AU} the saturation is achieved at a smaller value of λ .

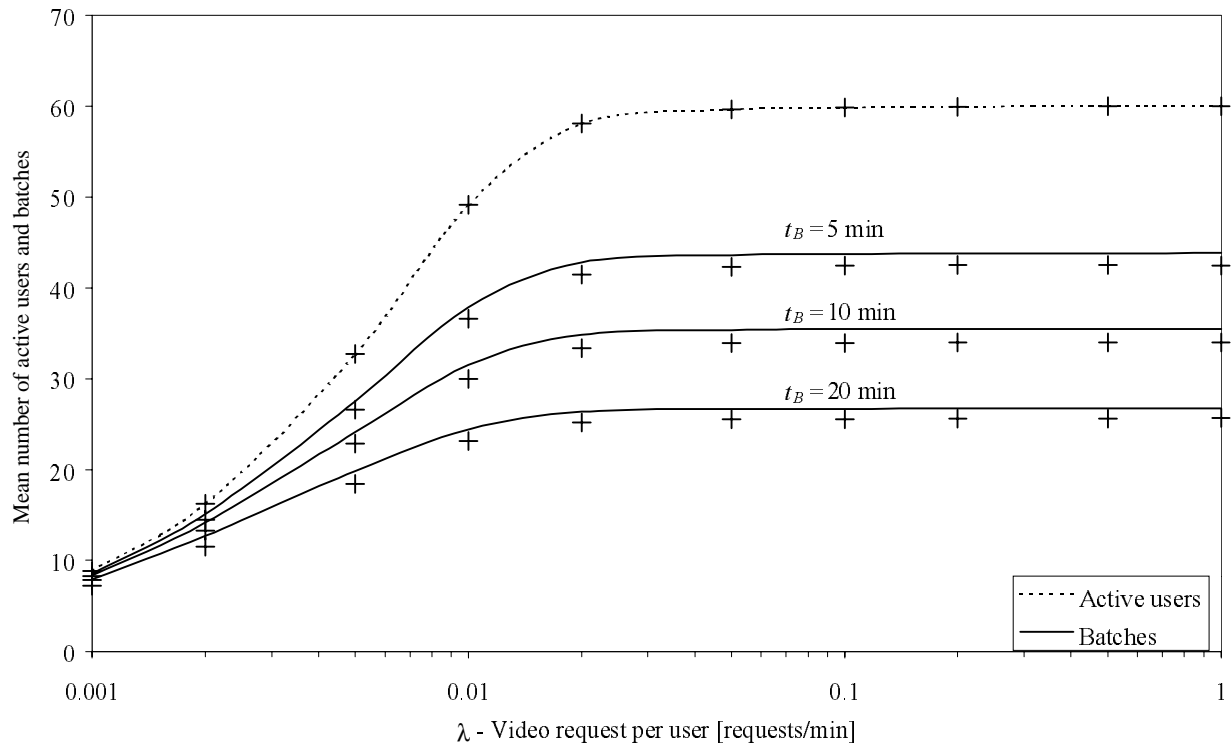


Figure 4. Mean number of active users and batches vs. λ ($N_{AU} = 60$ users)

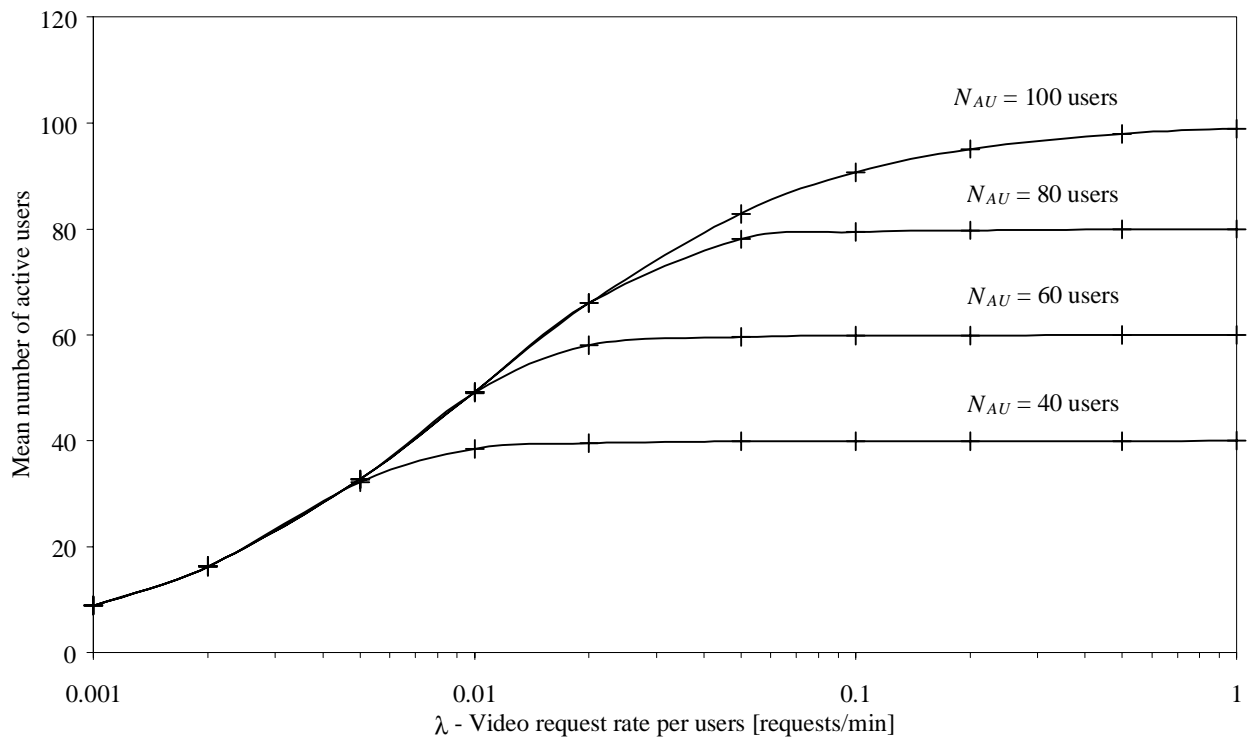


Figure 5. Mean number of active users vs. λ ($t_B = 10$ min)

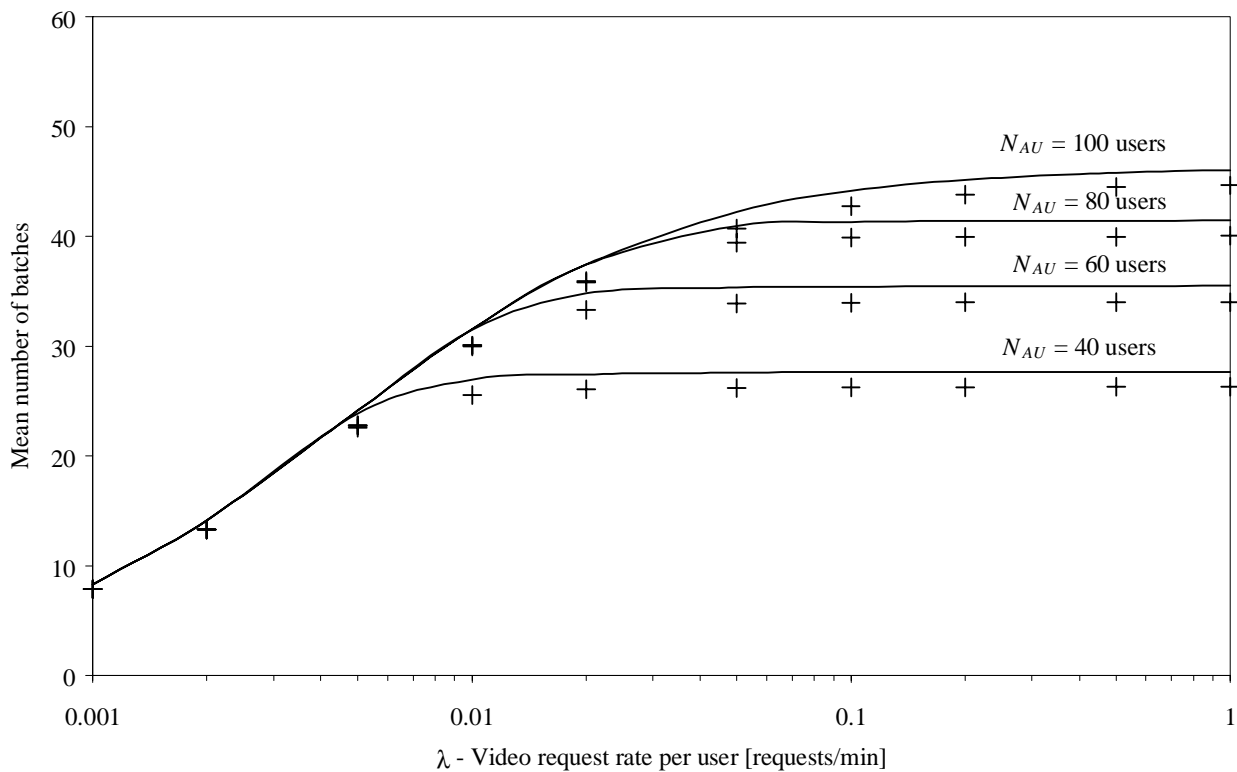


Figure 6. Mean number of batches vs. λ ($t_B = 10$ min)

Fig. 7 shows $E[n_B]$ as a function of N_{AU} ; we fix t_B to 10 minutes and observe $E[n_B]$ for different values of λ . For a fixed value of λ an increase in N_{AU} causes an increase in $E[n_B]$ even though t_B

does not change. This is due to the fact that more requests arrive to the system in the batching interval of fixed duration t_B ; such requests, eventually for different video objects, are then grouped in different batches, thus increasing the number of batches. As far as λ is concerned, let us observe that the curves with higher values of λ dispose all above those corresponding to lower values of λ , this is due to the fact that if λ grows, for fixed N_{AU} and t_B , there is a greater number of batches.

Fig. 8 shows $E[n_B]$ versus N_{AU} for different values of the parameter t_B . For a given value of N_{AU} and λ the mean number of batches decreases with the duration of t_B since a higher number of requests may arrive to the system for a wider t_B .

Figures 9 and 10 show $E[n_B]$ as a function of the batching interval. As expected, fixed the requests rate and the maximum number of active users, if t_B increases, the mean number of batches decreases. In fact, with higher values for t_B , we have more requests in the same batching interval and then a lower number of batches.

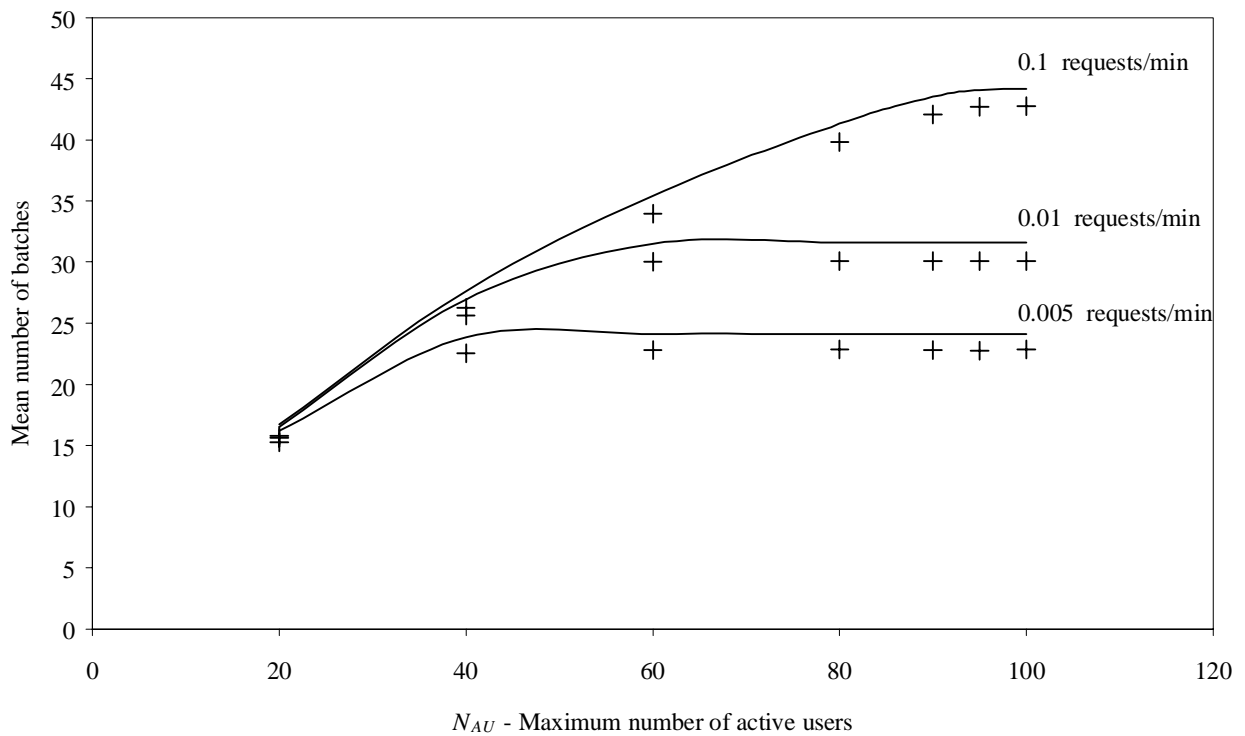


Figure 7. Mean number of batches vs. maximum number of active users ($t_B = 10$ min)

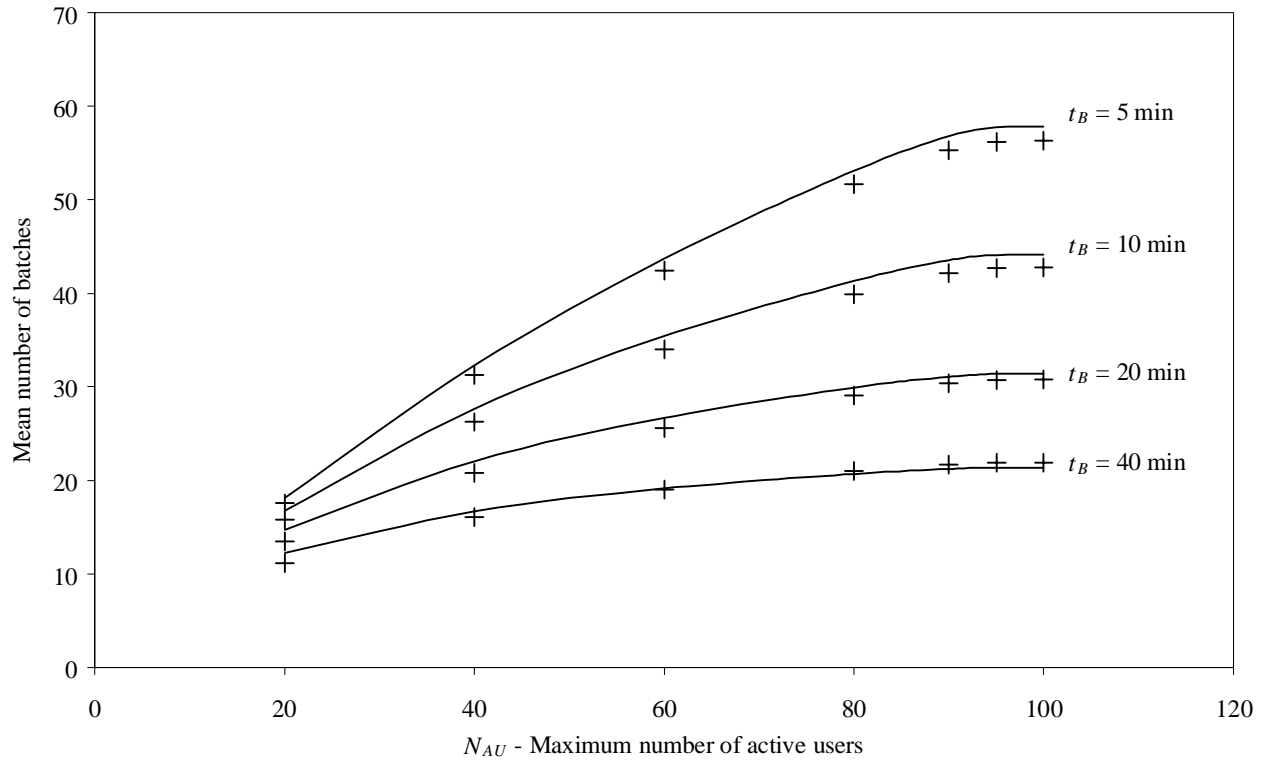


Figure 8. Mean number of batches vs. maximum number of active users ($\lambda=1$ requests/min)

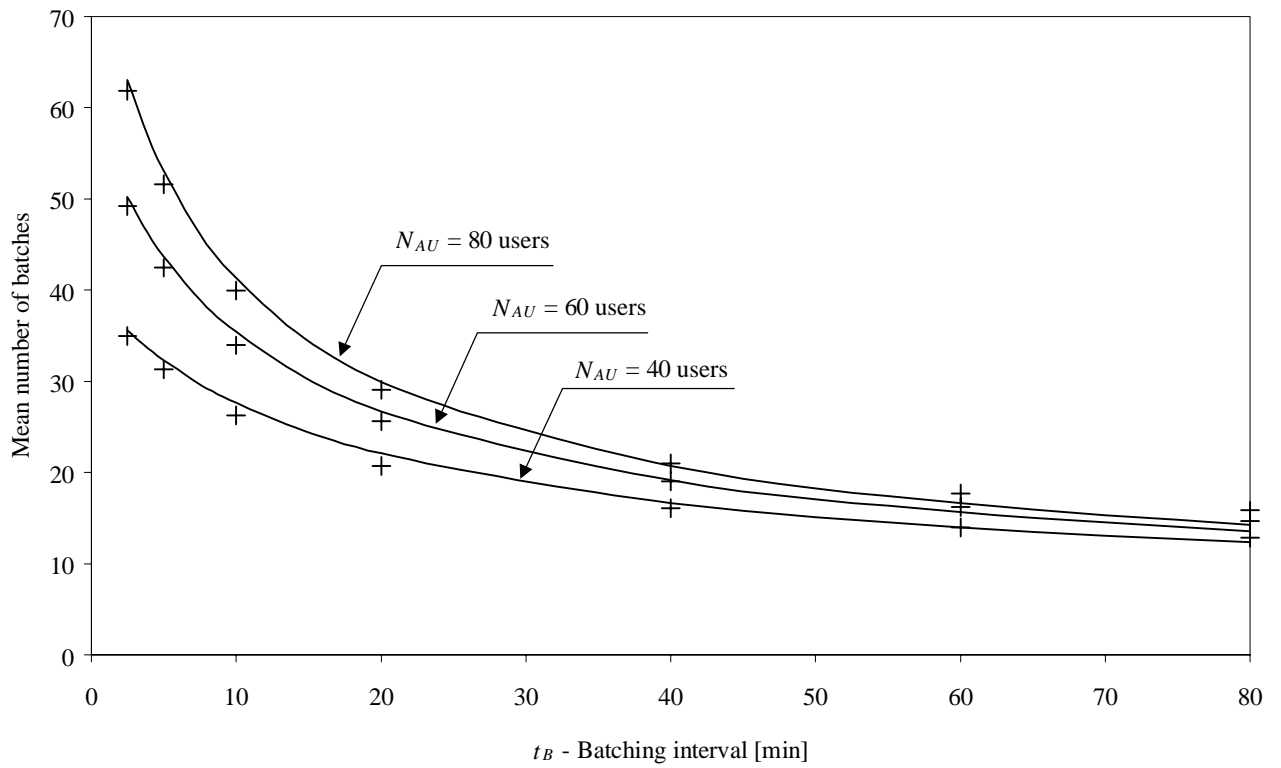


Figure 9. Mean number of batches vs. batching interval ($\lambda = 0.1$ requests/min)

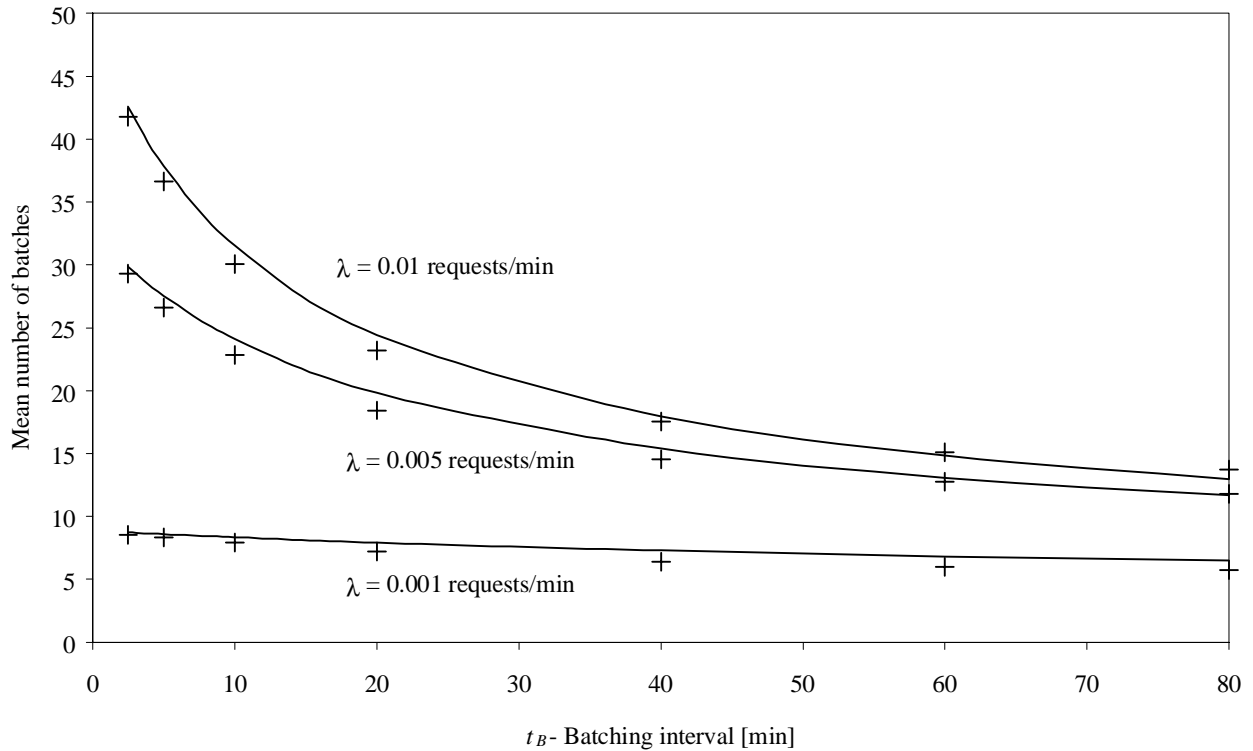


Figure 10. Mean number of batches vs. batching interval ($N_{AU} = 60$ users)

III.2 Percentage reduction of resource utilization

Let us observe that the percentage reduction of resource utilization, $R\%$, is meaningful when evaluating the bandwidth reduction in the case of shared bandwidth. Moreover, $R\%$ does not depend on the particular video object required by each batch and has a global meaning.

Figure 11 shows the behavior of $R\%$ versus λ for several values of N_{AU} and $t_B=10$ min. $R\%$ grows for higher values of N_{AU} , anyway such value strictly depends on the physical resources used to implement the system.

In Fig. 12 $R\%$ is reported as a function of λ with parameter t_B for $N_{AU}=60$. $R\%$ increases for higher value of t_B that means more main memory in case of buffering or larger response time for delivery video objects in case of batching.

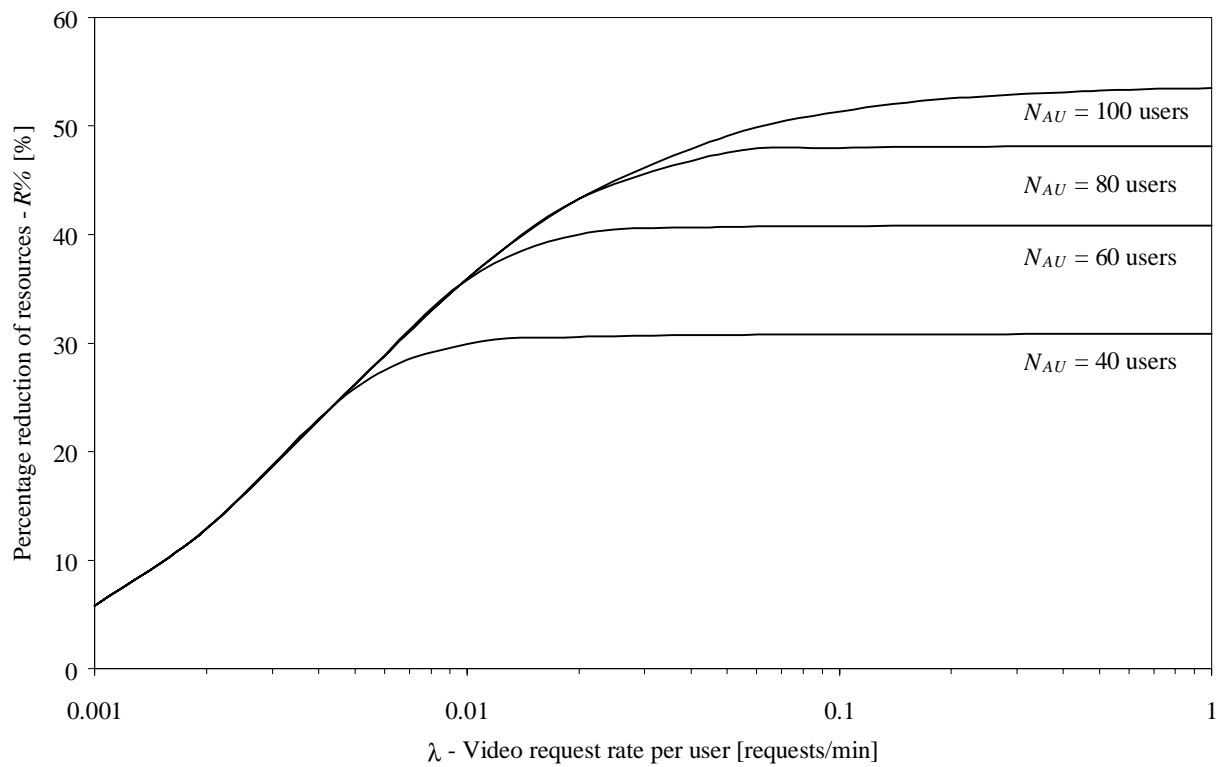


Figure 11. Percentage reduction of resources vs. λ ($t_B = 10$ min)

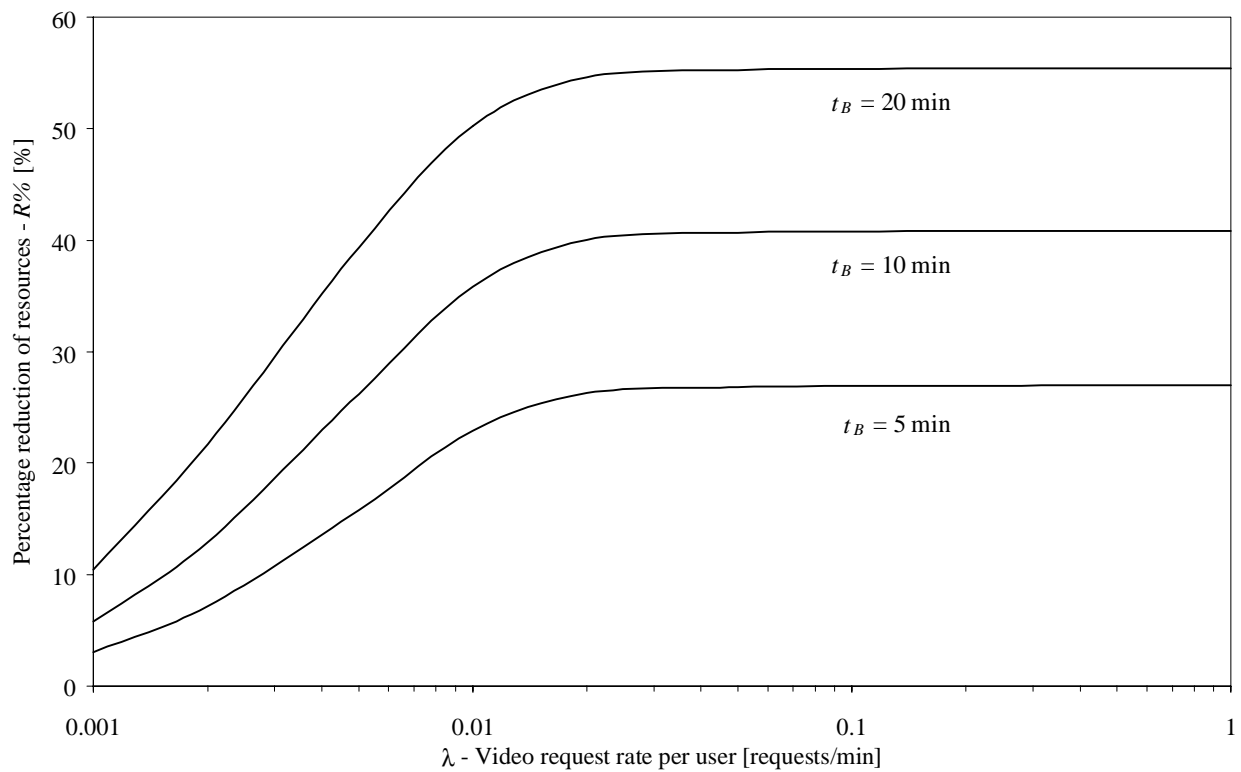


Figure 12. Percentage reduction of resources vs. λ ($N_{AU} = 60$ users)

III.3 Unsuccessful video request probability

Figure 13 shows the probability P_U that a customer request is refused, as a function of the arrival rate λ for varying values of N_{AU} . For small values of λ we note that P_U keeps close to zero; gradually, for increasing values of λ , P_U grows and reaches the maximum value for high values of λ . Such behavior is due to the fact that for small λ there is a small number of active users then a new video request is likely accepted. When λ grows, the number of users in the active state arises then the blocking probability is higher. The smaller N_{AU} the smaller is the value of λ where P_U becomes meaningful, such result is intuitively justified by a reduced capability of the system to satisfy customer requests.

In Fig. 14 P_U is function of N_{AU} ; the block has a high probability for small values of N_{AU} and decreases for increasing values of N_{AU} .

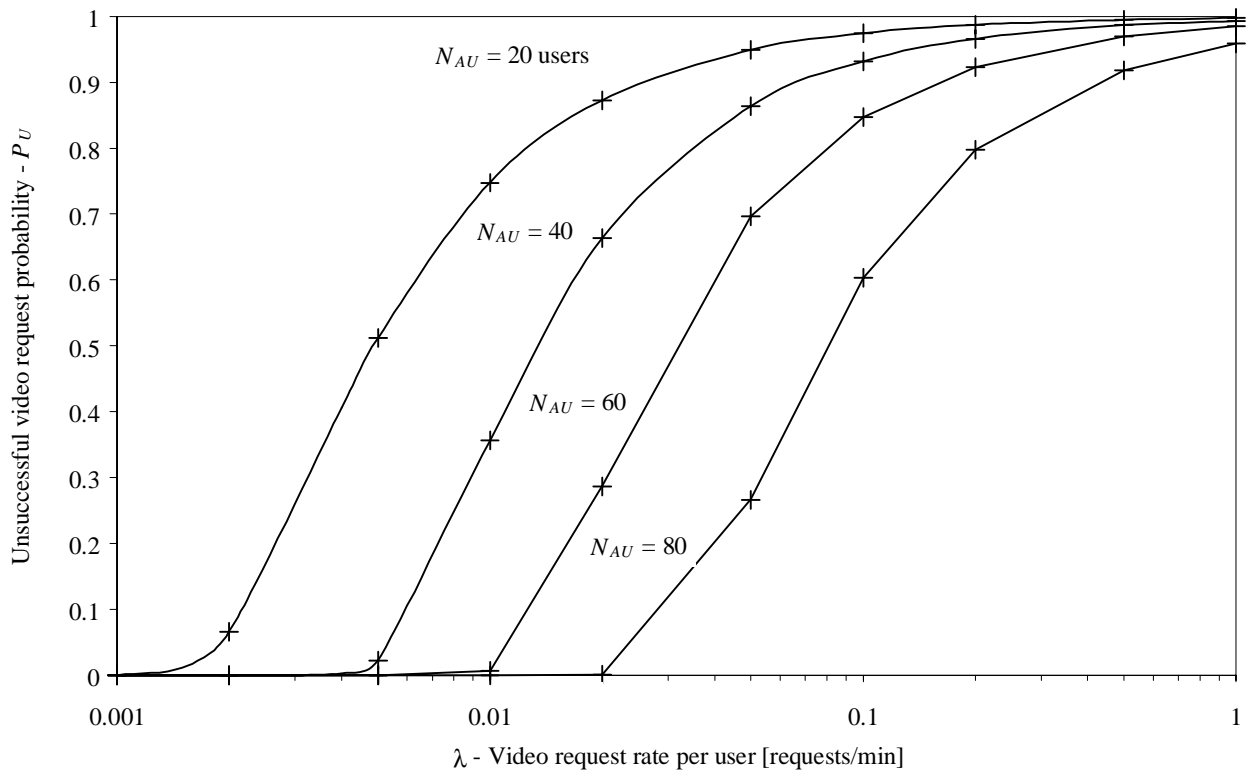


Figure 13. Unsuccessful video request probability vs. λ

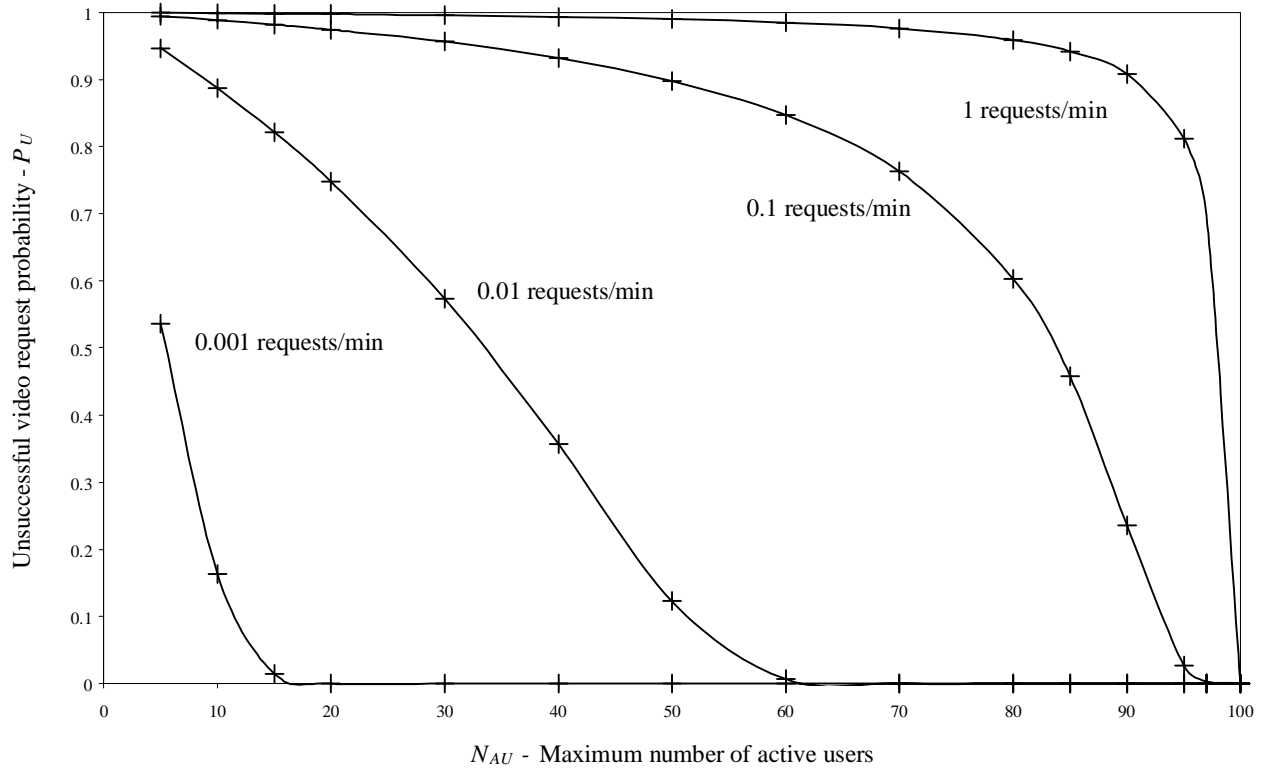


Figure 14. Unsuccessful video request probability vs. N_{AU}

In the architecture design, we can use the previous figures to choose the values of the system parameters. For example, with a fixed value of the number of users and of the video requests rate, from Fig. 14 we obtain the minimum value for N_{AU} ensuring the system works with a given unsuccessful probability; thus, fixed a batching interval, from a graph similar to Fig. 12, we can obtain the percentage reduction of system resources.

III.4 Simulation environment

The simulator is an event oriented simulation program written in the C language using the libraries of the SMPL simulation language [16].

Two main events characterize the system behavior, both describing the transitions of the user requests between the two service centers. Event 1 occurs when a user leaves the service center 1 to reach the service center 2 in the queuing network, this corresponds to a transition from the idle state to the active state. After the service, the user joins the service center 1 and returns to the idle state, thus event 2 occurs. In case of lack of resources of the center 2, the request is not accepted and the user makes a new attempt after a new round of service in center 1. The system clock advances when one of these two events occurs.

Interarrival times between two video requests for the same user are generated randomly by suitable exponential distributions. Users require videos following the request probability shown in Table I.

Through simulation we measure the mean number of active users in the system, the mean number of batches and the unsuccessful video request probability as follows:

$$E[n_{(2)}] = \frac{1}{T_{\max}} \int_0^{T_{\max}} n_{users}(t) dt ; \quad E[n_B] = \frac{1}{T_{\max}} \int_0^{T_{\max}} n_{batches}(t) dt ; \quad P_U = \frac{N_{Vref}}{N_{Vreq}} \quad (25)$$

where $n_{users}(t)$ and $n_{batches}(t)$ are respectively the number of users in the active state and the number of batches at the time t , and N_{VRef} is the number of user refused requests during the simulation, and N_{VReq} denotes the number of attempted requests of users over the simulation duration, that is T_{max} .

IV Conclusions

In this paper we have developed a simple analytical model, validated by simulation, to study the performance of a multimedia on-demand system with resource sharing, exploiting batching and buffering techniques.

As we can see from the obtained results, high values for the batching interval give high performance, in terms of reduction of resources requirement, e.g., transmission bandwidth, I/O bandwidth for video servers, etc. But this reduction causes an increase of the waiting time for each user with the batching technique or an increase of memory required to store the video frames with the buffering technique. Moreover, we have the maximum resources saving for high values of the video requests rate, but, in this case, the system operates with high unsuccessful probability. The developed model allows a good system design providing a tradeoff among these contrasting needs.

Appendix A

Let us consider a Poisson process with rate λ ; let t_1, t_2, \dots, t_j be the arrival instants. Now, we evaluate the probability $P(t_q - t_p \leq t_B, t_{q+1} - t_p > t_B)$ with $t_1 \leq t_p \leq t_q \leq t_{j-1}$. We have:

$$P(t_q - t_p \leq t_B, t_{q+1} - t_p > t_B) = P(t_{q+1} - t_q + t_q - t_p > t_B, t_q - t_p \leq t_B) = P(t_{q+1} - t_q + \tau > t_B, \tau \leq t_B). \quad (A.1)$$

Considering the total probability theorem and the memoryless property of Poisson process, observing that τ is an Erlang random variable with parameters λ and $q-p$, (A.1) becomes:

$$\int_0^{t_B} P(t_{q+1} - t_q + \tau > t_B | \tau = x) P(\tau = x) dx = \int_0^{t_B} P(t_{q+1} - t_q > t_B - x) \frac{e^{-\lambda x} \lambda^{q-p} x^{q-p-1}}{(q-p-1)!} dx. \quad (A.2)$$

The first term in the integral is the probability that in the time interval $t_B - x$ there are no arrivals,

i.e., $e^{-\lambda(t_B-x)}$ by Poisson distribution, thus:

$$P(t_q-t_p \leq t_B, t_{q+1}-t_p > t_B) = \int_0^{t_B} e^{-\lambda(t_B-x)} \frac{e^{-\lambda x} \lambda^{q-p} x^{q-p-1}}{(q-p-1)!} dx = \frac{e^{-\lambda t_B} (\lambda t_B)^{q-p}}{(q-p)!}. \quad (\text{A.3})$$

Appendix B

Here, we show a simple algorithm to evaluate the expression (15) with low computational costs. Since the minimum dimension for each batch is one (i.e., each $\alpha_k \geq 1$ where α_k is still the number of users in the k^{th} batch) we can observe that it is possible to rewrite (15) as:

$$P(n_{Bc}=i / n_c=j) = \sum_{k=i-1}^{j-1} \left(\sum_{\alpha_1+\dots+\alpha_{i-1}=k} \Phi(\alpha_1)\dots\Phi(\alpha_{i-1}) \right) \Psi(\alpha_i = j-k) = \sum_{k=i-1}^{j-1} \tilde{P}(i-1 | k) \Psi(j-k) \quad (\text{B.1})$$

Fixed $j=2, \dots, N_{AU}$, it is possible to evaluate $\tilde{P}(i | j)$ with a recursive expression, in fact:

$$\tilde{P}(i | j) = \sum_{\alpha_1+\dots+\alpha_i=j} \Phi(\alpha_1)\dots\Phi(\alpha_i) = \sum_{k=i-1}^{j-1} \left(\sum_{\alpha_1+\dots+\alpha_{i-1}=k} \Phi(\alpha_1)\dots\Phi(\alpha_{i-1}) \right) \Phi(\alpha_i = j-k) \quad (\text{B.2})$$

that is:

$$\tilde{P}(i | j) = \sum_{k=i-1}^{j-1} \tilde{P}(i-1 | k) \Phi(j-k) \quad (\text{B.3})$$

with $i=2, \dots, j$ and the initial condition:

$$\tilde{P}(1 | j) = \Phi(j) \quad (\text{B.4})$$

Now, we can build a simple algorithm to evaluate each $P(n_{Bc}=i / n_c=j)$ for $i=1, \dots, N_{AU}$ and $j = i, \dots, N_{AU}$:

- *Step 1:* evaluation of each $\Phi(i)$ and $\Psi(i)$ for $i=1, \dots, N_{AU}-1$;
- *Step 2:* evaluation of initial conditions $\tilde{P}(1 | j) = \Phi(j)$ for $j = 1, \dots, N_{AU}$;
- *Step 3:* for $i = 2, \dots, N_{AU}$

for $j = i, \dots, N_{AU}$

evaluation of $P(n_{Bc}=i / n_c=j)$ and $\tilde{P}(i | j)$.

In this algorithm, all loops require $O(N_{AU}^2)$ iterations.

References

- [1] V.O.K.Li, W.Liao, X.Qiu and E.W.M.Wong, "Performance Model of Interactive Video-on-Demand Systems", *IEEE JSAC*, Vol.14, No.6, August 1996, pp. 1099-1109.
- [2] D. J. Gemmell, H. M. Vin, D. D. Kandhlur, P. Venkhat Rangan, "Multimedia storage servers: a tutorial and survey", *IEEE Computer*, Vol. 28, No. 5, May 1995, pp. 40-49.
- [3] J.P. Naussbaumer, B.V. Patel, "Network requirement for Interactive Video on Demand", *IEEE JSAC*, Vol. 13, No. 5, June 1995, pp. 779-787.
- [4] S.W. Lau, J.C.S. Lui, L. Golubchik, "Merging video streams in a multimedia storage server: complexity and heuristics", *Multimedia Systems*, Vol. 6, 1998, pp. 29-42.
- [5] L. Golubchik, J.C.S. Lui, R. Muntz, "Reducing I/O demand in Video on Demand storage servers", *Proceedings of the ACM SIGMETRICS/PERFORMANCE '95 Conference*, Ottawa, Canada.
- [6] K.C. Almeroth, M.H. Ammar, "The use of Multicast delivery to provide a scalable and interactive Video-on-Demand service", *IEEE JSAC*, Vol. 14, No. 6, August 1996, pp. 1110-1122.
- [7] H. Shachnai, P.S. Yu, "On analytic modeling of multimedia batching schemes", *Performance Evaluation*, Vol. 33, 1998.
- [8] M. Kamath, D. Towsley, K. Ramamritham, "Continuous media sharing in multimedia database systems", *Fourth International Conference on Database Systems for Advanced Applications (DASFAA'95)*, Singapore, pp. 79-86.
- [9] S. Sen, L.Gao, J.Rexford, D.Towsley, "Optimal Patching Schemes for Efficient Multimedia Streaming", University of Massachusetts CMPSCI, Tech.Rep. 99-22.
- [10] L.Gao, D.Towsley, "Supplying instantaneous Video-on Demand services using controlled multicast" *IEEE Multimedia Computing and Systems*, June 1999.
- [11] S. Sen, J.Rexford, D.Towsley, "Proxy Prefix Caching for Multimedia Streams", University of Massachusetts CMPSCI, Tech.Rep. 98-27.
- [12] L. Kleinrock, *Queueing systems, Volume I: Theory*, Wiley & Sons, 1975.
- [13] S. Lavenberg, editor, *Computer performance modeling handbook*, Academic Press, 1983.
- [14] N. M. Van Dijk, *Queueing networks and product forms. A system approach*, Wiley & Sons, 1993.
- [15] A. L. Garcia, *Probability and random processes for electrical engineering*, Addison Wesley, 1989.
- [16] M. H. Dougall, *Simulating computer systems: techniques and tools*, MIT Press, 1987.