# A MULTI-LEVEL SATELLITE CHANNEL ALLOCATION ALGORITHM FOR REAL-TIME VBR DATA [1]

Nedo Celandroni, Erina Ferro, Francesco Potortì

CNUCE, Institute of National Research Council (C.N.R.)
C.N.R. Pisa Research Area
Via Alfieri 1, 56010 Ghezzano (Pisa), Italy
Phone: +39-050-3152988/ 3153070/ 3153058 - Fax: +39-050-3138091/3138092
E-mail: {N.Celandroni, E.Ferro, F.Potorti}@cnuce.cnr.it

## Abstract

Integrated services over a shared satellite channel need a MAC protocol that is able to deal with real-time traffic without substantially affecting the efficiency of the shared medium. Because of its bursty nature, MPEG coded VBR video transmission is one of the most challenging real-time applications. In this paper we analyze the statistical performance of a simple allocation method for VBR traffic by using an accurate statistical model of an MPEG-2 VBR coded movie, whose traffic exhibits a peak to mean ratio of about 4. The results were obtained both analytically and by simulation, and show that our method is suitable for transmitting non-interactive video and best-effort traffic on the same satellite link. The efficiency of the proposed method shows no dependency on the statistical properties of the input traffic.

**Keywords**: satellite, VBR, MPEG-2, bursty traffic, packet delay, channel efficiency, allocation levels

## 1. Introduction

The transmission of real-time, multimedia data streams on a satellite channel together with lower-priority (non real-time) data is a challenge for system designers. In fact, the medium access control (MAC) protocol used for the satellite link must be able to guarantee both high link utilization and low delay transmission for the variable bit rate (VBR), real-time data. This can be accomplished by filling with non real-time data the part of channel bandwidth unused by the VBR data stream. By non real-time data we mean both traditional EDP batch data transmission, and best-effort internet data such as mail, news, or web browsing, which do not

[1] Work supported by the Italian National Reserach Council (C.N.R.) under the 5% Multimedia

require any hard guarantees on either transmission deadlines or available bandwidth. On the other hand, audio and video data do have strict deadline and bandwidth requirements. In particular, real-time digital video needs a high transmission bandwidth even after compression, it must be sent with a minimal delay, and it cannot tolerate a high error rate. For example, "high quality" MPEG-2 coded video data requires an average transmission capacity in the range from 4 to 13 Mbps. Such a high data rate often puts a strain on existing local area networks (LANs), while most of the existing wide area networks (WANs) would not even be able to transmit real time data at such a high rate. Delay is critical in real-time video transmissions because video frames must be presented to the viewer at a constant rate, typically in the range of 24 to 30 frames per second. Therefore, real time coded video cannot tolerate large variations in transmission delay such as those caused by most existing networks.

In fact, dedicated transmission lines are a good choice for constant bit rate (CBR) traffic, but are underused for VBR traffic, especially MPEG-2 VBR traffic for which the peak/mean ratio computed over a group of pictures (GOP) may be as high as 6. On the other hand, current general-purpose networks are designed for best-effort traffic. Much study and work is under way to design general purpose Internet protocols that can be used for both best-effort and real-time traffic [1-4]. In order to meet their commitments, these protocols need underlying MAC protocols that can provide bandwidth and/or delay guarantees. This paper proposes a TDMA (Time Division Multiple Access) MAC protocol for shared satellite channels which is able to provide good channel efficiency for mixed traffic, while still guaranteeing delay and bandwidth for MPEG-2 VBR traffic.

The paper is organized as follows. Section 2 illustrates the network scenario; Section 3 describes the proposed allocation algorithm; Section 4 deals with the traffic generator used to feed the simulator; Section 5 contains the details of the simulation environment, and Section 6 presents both the simulation and the analytical results. Conclusions are drawn in the last Section.

## 2. Network scenario

Let us assume that we have a satellite network composed by one geostationary satellite and multiple earth stations sharing a common satellite link. Each earth station acts as a concentrator or gateway, connecting one or more terrestrial networks to the satellite network, which is capable of exchanging both real-time and best-effort traffic. A centralized or a distributed TDMA MAC protocol allows the satellite network to be shared. In the centralized case, a master station is responsible for receiving the allocation requests and granting a channel share to

---

Programme.

the requesting earth stations. In the distributed case, each station receives the allocation requests of all the other stations, so a fully meshed network is needed. All the stations run the same algorithm to decide how the requests should be handled and what the allocation plan for transmission should be. In the case of centralized allocation, the master builds a *transmission plan* and broadcasts it, so that every station knows when it is scheduled to transmit. In the distributed case, each station computes the transmission plan by itself, and schedules its own transmissions accordingly. In this case, great care must be taken to ensure consistency between the stations' ideas of what the transmission plan should be [5-6].

In both cases there is an algorithm that computes the transmission scheduling for each station: whether this is a centralized or distributed algorithm, we say that a station issues a request to the *channel dispatcher* and receives an allocation from it. Only after the channel dispatcher has granted the allocation will the earth station use the shared satellite channel. This means that, when an earth station realizes that the current allocation is not sufficient for the traffic coming from the terrestrial network, it must ask the channel dispatcher for a larger share of the channel bandwidth, and it queues the incoming traffic while waiting for its request to be granted.

Requests to the channel dispatcher are made separately for real-time and non real-time traffic, and the network uses different algorithms to allocate bandwidth to each station in order to satisfy the requests. The bandwidth allocation algorithm used for non real-time traffic is not described in this paper, but many can be found in the literature [7-12].

## 3. The n-levels assignment algorithm

There is a non-negligible delay from the moment a request is made to the channel dispatcher and when the allocation is granted to the station, due to the propagation delay. This delay is one round trip time (RTT, about 250 ms for geostationary satellites) for networks with distributed control, and two round trip times for centralized control, assuming a single-hop satellite network. Since no distributed control geostationary networks are currently in use, hereafter we will assume a 500 ms delay from the station's request for bandwidth and the relative allocation.

Because of the allocation delay, the station has to estimate in advance what allocation it will need 500 ms later. Since this estimation is generally affected by error, the allocated space may be too large or too small. When the station has unused real-time allocation, because the real-time traffic coming from the terrestrial network is less than the satellite link share available, it can use this allocation for sending non real-time data, if any is queued at the station. The station can thus avoid wasting bandwidth that has been granted to it. An application, which is able to exploit an efficient algorithm for allocating satellite bandwidth on demand to MPEG sources, is discussed, for example, in [13].

In [14] we presented a demand-assignment (DA) centralized control allocation algorithm for real-time traffic based on two levels of bandwidth allocation for VBR traffic (V2L-DA/TDMA). The algorithm guarantees the peak bandwidth (*P*), while maintaining a good efficiency in the overall channel allocation. In fact, the throughput of a VBR application is often several times lower than its peak value, and this leads to an inefficient use of the channel bandwidth if the peak rate is always allocated. In V2L, when the throughput of a VBR transmission is below a given threshold *R*, the transmitting station gives up the excess allocation *P* - *R*, while keeping it booked, and retains only an allocation equal to *R*. The sum of all bookings in the satellite network cannot exceed the link capacity. This requirement ensures that any station can reclaim its booked bandwidth *P* at any moment. When a station releases its excess allocation, the channel dispatcher can allocate it to stations that request it for non real-time use. If the VBR data throughput is less than *R*, the unused space is devoted to the non real-time traffic of the sending station itself. As soon as the throughput of the VBR encoder exceeds *R*, the station makes a request, and the channel dispatcher allocates all the bandwidth *P* that the station had booked. In [14] we considered the trace-driven transmission of the movie “The sheltering sky” produced by an MPEG-1 encoder. The optimal bandwidth allocation for this VBR video application was obtained by setting *R* to about 40% of the booked peak bandwidth *P*, which, in turn, was set to 5/8 of the GOP peak rate. Considering that for about two thirds of the time the source bit rate is below R, it follows that about 40% of the bandwidth booked by a VBR video application can be shared among all the stations for transmitting non real-time data.

In this paper we present an extension of the V2L algorithm to *n* levels of allocations, which we call VnL, and we demonstrate that the efficiency of the channel utilization increases with the number of levels. The performance of the allocation algorithm is simulated by using a synthetic MPEG-2 generator, and both delay and channel usage efficiency results are presented.

Three parameters (*Amin*, *Amax*, *nlev*) define the minimum throughput, maximum throughput (booking) required by a real-time application, and the number of allocation levels, respectively. The channel dispatcher accepts a new allocation request for real-time traffic only if the sum of the new and the outstanding bookings does not exceed the percentage of the channel bandwidth that is dedicated to the real-time traffic. This threshold can be tuned to avoid starvation of non real-time traffic. However, with the use of many allocation levels, this problem is not likely to be an issue, because most of the time the space allocated for real-time traffic is much less than the space booked. As in V2L, any channel space allocated to a station for real-time data and not used can be used by that same station to send its non real-time traffic. The relationship between *Amin* and *Amax* depends on the type of real-time application that generates the request. If

*Amin* = *Amax*, the request comes from a CBR application. In the following, we study the case where *Amin* is less than *Amax,* i.e. the request relates to a VBR application.

As depicted in Figure 1, the *nlev* allocation levels are equally spaced in the range [*Amin*, *Amax*].

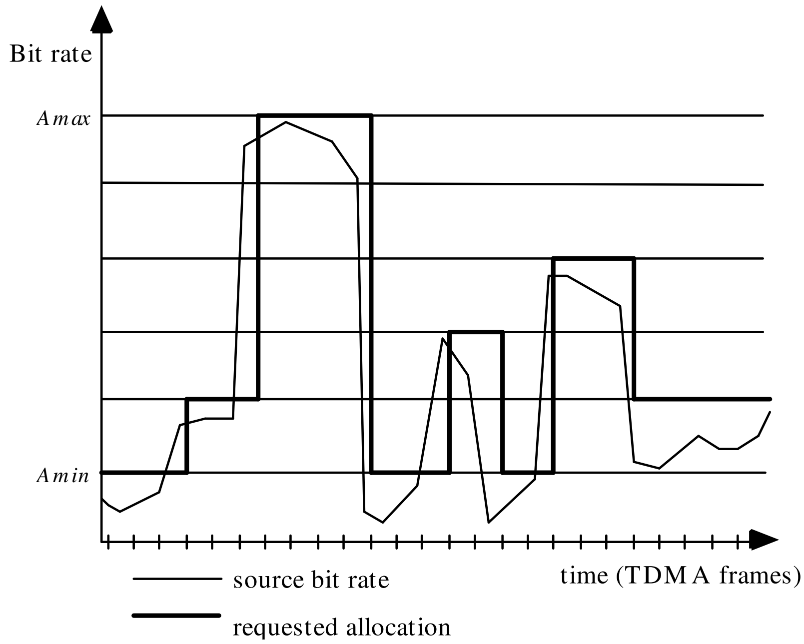



*Figure 1. Allocation levels for input traffic with varying throughput*

For each VBR flow entering a station, the station books a bandwidth *Amax*. Once the allocation has been granted, the station keeps measuring the throughput of the flow and requests an allocation equal to one of the *nlev* allocation levels. As in V2L, the bandwidth booked but not allocated is managed by the channel dispatcher to satisfy requests for non real-time traffic allocation.

The earth stations measure the input traffic at discrete instants of time. For a TDMA system, like the one we considered, a TDMA frame is the natural unit of measurement, assuming that each station receives an allocation in each frame. In order to compute the correct allocation level to request each station keeps two counters for each VBR flow, which we call a positive and negative virtual queue, *pvq* and *nvq*, respectively. The *pvq* keeps track of the volume of data that would be queued at the station if the requested allocation were granted immediately. The *nvq* is the cumulative unused allocation space that would be wasted if the allocation were one level below the requested one.

Let us detail the algorithm for computing the appropriate *nlev* value. Let $I_i$ be the volume of input traffic to a station for a VBR flow during frame *i*, $A_{i\text{-}1}$ the last requested allocation level, and $pvq_i$ and $nvq_i$ the virtual queues for the current frame. At frame $i$, the virtual queues are updated as follows:

$$Astep = (Amax-Amin) / (nlev-1);$$

$$pvq_i = \max (0, pvq_{i-1} + I_i - A_{i-1});$$

$$nvq_i = \min (0, nvq_{i-1} + I_i - (A_{i-1} - Astep)) \qquad \text{if } A_{i-1} > Amin, \text{ else } 0.$$

When the data throughput is between the current allocation level and the one below it, both virtual queues are 0; otherwise one and only one of the two virtual queues is different from 0. If *pvq* is positive, then a request is made for a higher allocation level, while a request is made for a lower allocation level if *nvq* is negative. $A_i$ is the value of the allocation request for the current frame. To be more precise:

$$A_i = \min (Amax, A_{i-1} + [\mathrm{ceil} (pvq_i / (T\ Astep)) + \mathrm{floor} (nvq_i / (T\ Astep))]\ Astep)$$

where $T$ is the time interval between successive requests, *floor* gives the greatest integer not greater than its argument, and *ceil* gives the smallest integer that is not smaller than its argument. All the requests are made on the basis of the last allocation level requested, not on what is currently granted by the channel dispatcher. This means that queues build up at the station when the input traffic increases, while unused allocation will be granted to the station when the input traffic decreases.

Note that a real implementation of this algorithm would have to account for possible communication errors between the earth station and the channel dispatcher. In fact, VnL as described has no feedback mechanism that monitors the input queue, but only considers what we have called virtual queues, that is, what the input queue would be if the granted allocation were equal to the requested one.

A simple and effective method for making the algorithm resilient to control data errors is to verify that the requested allocation has in fact been granted after the allocation delay has elapsed. The allocation delay depends on whether the satellite network runs with centralized or distributed control and on the particular implementation of the access control, but it can safely be assumed that it is known a priori. If the requested allocation has not been granted when expected, because of a communication problem, then the virtual queues are adjusted according to the allocation that has in fact been granted. This simple process ensures that subsequent requests will take into account the possible excess traffic queued at the station. This solution is not a workaround for a problem that could be solved more exactly or cleanly, in fact there is no way for the sending station to know of such errors until one allocation delay has elapsed since the allocation request was issued.

## 4. Synthetic generation of MPEG-2 traffic

In order to reliably evaluate the performance of VnL, we fed it with a synthetically generated MPEG-2 compressed movie. We assume that a GOP is made up of twelve 40ms long frames, and that the output of the MPEG-2 coder is smoothed at the GOP level, i.e., that after the coder there is a 480ms pre-buffering before transmission on the network. This is a reasonable hypothesis under the assumption that satellite systems are a good multicasting or broadcasting medium for non-interactive real-time data, but not for interactive live video. In fact, the overall end-to-end delay for the visualization of live interactive data must account for the coding and decoding delays (at least four video frames), the packetization delay, and network delays caused by latency and buffering. Since network latency is 250ms for the satellite link only, the overall end-to-end delay is in the order of 500ms, which is not good for quick interactive applications (e.g. on line games). Adding a further 480ms for the prebuffering does not pose a problem for the transmission of prerecorded images, and alleviates the strain put on the network by the wildly varying throughput of an MPEG-2 VBR flow. Notice, however, that the VnL algorithm itself doesn't require the VBR data stream to be prebuffered, as it makes no assumptions about the characteristics of the traffic.

Using synthetically generated traffic instead of the real traces makes it possible to do a rigorous statistical analysis of the results, by evaluating the confidence intervals after doing independent replications of the simulation suite. This was particularly important for the quantile computations, for which a significant number of independent replication was necessary to obtain the desired accuracy.

The basis for the computation of the parameters of the VBR traffic generator was the same movie as used in [14]. The model of the MPEG video source used is sketched in [14], and described in [15, 16] in more detail. The generator uses a bidimensional Markov chain $\{M_j, H_j \mid 0 \leq j \leq 7\}$, where $H_j$ is the $j$th GOP size, and $M_j$ is the status of a low frequency process modulating the $j$th GOP size. No attempt is made to characterize the per-picture throughput variability: the time granularity of the model is limited to the GOP. $\{H_k | k \geq 0\}$ describes the bit rate per group of an MPEG encoder.

To represent the low-frequency component of the source, a modulating process $\{L_k | k \geq 0\}$ was included in the model as well $(L_k \in \{0, 1, 2, \ldots, M-1\})$. In the trajectories of the Markov chain, the $H_k$ value frequently changes (every few GOPs, on average) while the $L_k$ value changes on a much longer time scale (about 70-100 GOPs).

The transition probabilities of the Markov chain $\{L_k, H_k \mid k \geq 0\}$ are estimated from the trace of the movie by applying the procedure presented in [17]. The Markov chain captures both the short range dependency of the VBR flow, which lasts for a small number of GOPs (15 seconds), and long range dependencies, which last for thousands of GOPs (10-20 minutes), typical of the GOP sequences generated by MPEG coders.
Specifically, the model used for results presented in this paper is obtained with parameters M=8 and N=8. States with H=5, 6, and 7 are very rare, i.e. they only occur a few times in our two-hour sequence, but never consecutively.
The accuracy of this model was investigated in [15, 16], where it is shown that both the qualitative (i.e. burstiness and the overall appearance of traces) and the statistical properties (maximum, minimum, average, standard deviation and autocorrelation function) of the generated GOP size sequence are very similar to those of the real trace. Figure 2 shows the probability density function of the generated throughput.

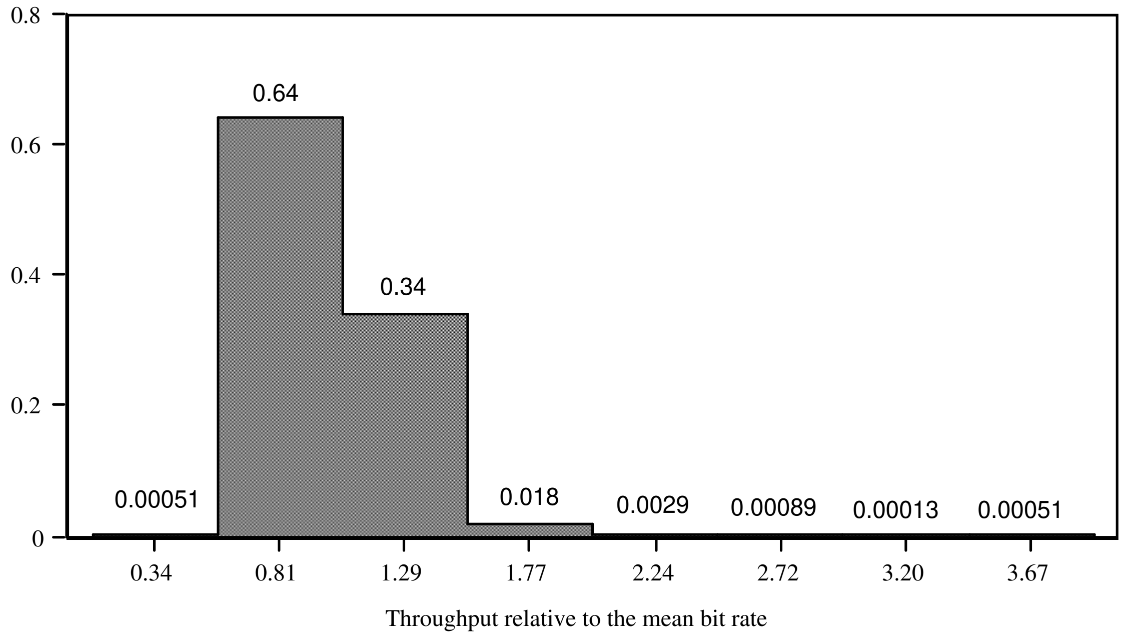



*Figure 2. Probability density function of the bit rate produced by the Markov generator*

## 5. Simulation environment and specifications

We run the simulations using FRACAS[(3)], a high speed, lightweight simulator useful for simulating framed channel allocation schemes [18]. It is written in C for maximum portability, and it is suitable both for comparing different satellite channel allocation policies and for tuning and performance evaluation. It includes some built-in allocation policies, selected from those available in the literature, and some traffic generators, both random and trace-driven. We included in FRACAS the VnL policy and the Markov generator described in the previous sections. In order to obtain statistically significant results, we also wrote a module, using the Python language, which implements independent replications by repeatedly calling FRACAS with different seeds for the random number generators, until the requested confidence interval for the results has been obtained.

Here are the most significant specifications of the simulation runs:

- a TDMA system is used, where the frame length is 20 ms, the virtual queues are probed once every frame, and an allocation request is issued at every frame;
- the VBR source is simulated by using the Markov generator described in the previous section, with a mean and a maximum throughput equal to 3 Mbps and 11.7 Mbps, respectively;
- a single traffic station is loaded with the VBR traffic generator. This is because loading more than one station with VBR traffic does not give any additional information on the system performance. In fact, once a request is granted, how a station utilizes its bandwidth does not influence the behaviour of the other stations. The number of stations loaded with VBR traffic only influences the probability that a request be granted or not.
- the statistics collected include mean unused space, maximum packet delay, and packet delay quantiles of 0.9, 0.99 and 0.999;
- the minimum allocation *Amin* has been varied between 1 and 5 Mbps, in steps of 0.1 Mbps, while the maximum allocation *Amax* lies in the range between 5 and 9.5 Mbps, in steps of 1.5 Mbps;
- the number of allocation levels *nlev* has been set to 2, 3, 4, 10 and 100; the 100 level case is a practical approximation of a continuous variation in the allocation level.

All the simulation results were obtained with a 95% confidence level. The size of the confidence intervals for the unused allocation were ±2%, while the confidence intervals for the delays were ±5%.

---

(3) FRAmed Channel Access Simulator, developed at CNUCE/C.N.R. Pisa (IT).

## 6. The results

Two statistics were used to evaluate the performance of the proposed method: the end-to-end packet delay, and the unused allocation space. The *end-to-end packet delay*, which was computed by simulation only, measures the time a packet takes to cross the satellite network from earth station to earth station. This delay is variable because of variable queuing delays. The minimum delay is set to 250 ms, which is the conventional round-trip time of a geostationary satellite. The other statistic considered, the *unused allocation space*, was computed both analytically and by simulation. It represents the satellite link share allocated to a station for transmitting VBR traffic and used by the station for transmitting its non-real time traffic, if any. This statistic is a measure of the efficiency of the allocation method, and should be made as small as possible.

The quantiles of the packet delay capture the dynamic behaviour of VnL, and show how the variability of the input traffic affects its end-to-end delay. The unused allocation is mostly a steady state characteristic. In fact, a similar characteristic was studied in [14], yielding the same results as those obtained for VnL when the number of levels is 2.

### *6.1 End-to-end packet delay*

The packet delay is essentially the sum of three addenda. The first one is simply the latency of the satellite link. The second is the allocation delay when switching between levels in response to variations in the throughput of the VBR flow entering the earth station. During the two round trip times between the request for a larger allocation and the relative authorization (which is always granted, because the bandwidth has been booked in advance), the traffic is enqueued at the station, and the queue is emptied only after the allocation delay. There is always this effect when going up levels, and it depends on the *Amin*, *Amax*, and *nlev* parameters.

The third addendum of delay depends on the insufficiency of the booked allocation, *Amax*, which we set up to a number of values ranging from 5 Mbps to 9.5 Mbps, lower than the peak VBR data throughput, which is 11.7 Mbps. This third effect, which decreases as *Amax* increases, disappears when the maximum allocation is equal to the peak throughput of the input traffic, and it is independent both of the minimum allocation *Amin* and of the number of levels *nlev*. In order to eliminate this effect, and to examine the switching delay more closely, we also made some simulation runs with a maximum allocation of 12 Mbps, which is greater than the peak.

Figure 3 shows the relationship between the data delay and the maximum allocation when the minimum allocation is 3 Mbps and two levels are used. Indeed, the figure is exactly the same whatever the number of levels is, up to 100 levels, thus confirming that the delay is practically independent of the number of levels. In 99% of cases a maximum allocation of 8 Mbps is sufficient to fit the input traffic, since the data delay is about the same as that obtained with a maximum allocation of 12 Mbps, while a maximum allocation of 6.5 Mbps, or lower, satisfies the input traffic only in 90% of cases.

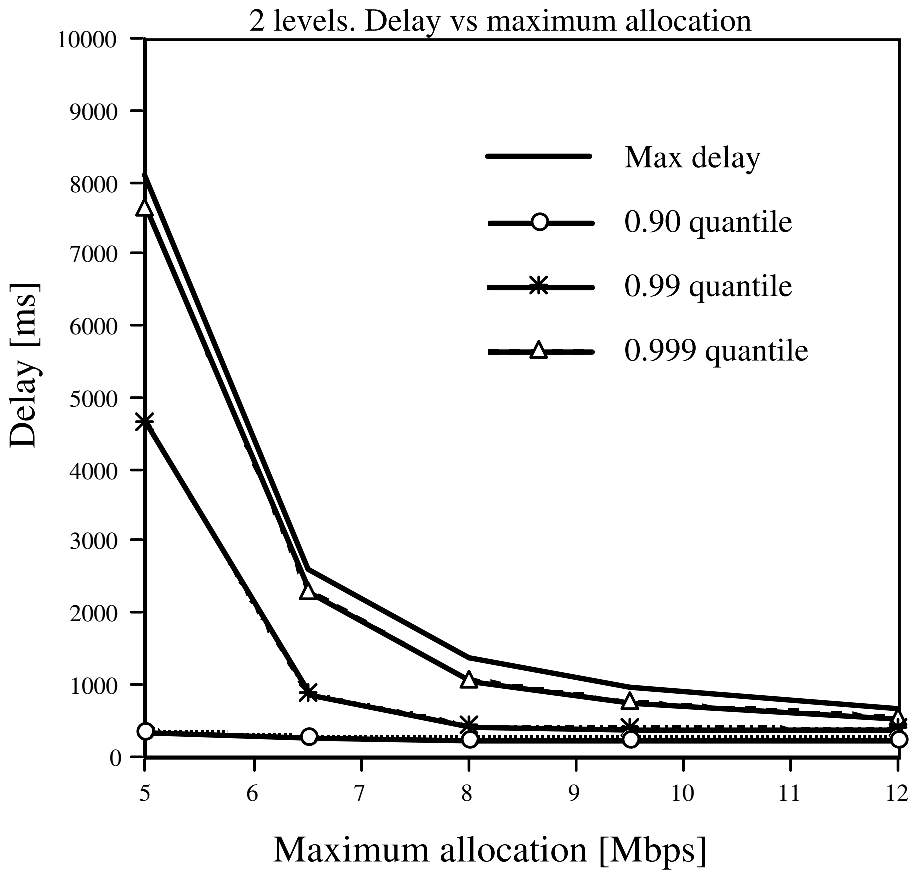



*Figure 3. Data delay vs. maximum allocation. Two allocation levels with a minimum allocation equal to 3 Mbps. Same result for any allocation level.*

Figure 4 plots the data delay versus the minimum allocation for 100 levels. The runs were made with a maximum allocation of 5, 6.5, 8, 9.5, and 12 Mbps, with the same confidence intervals as the previous results. We present here only the results for 100 allocation levels, as the plots for different numbers of levels are practically identical. The data delay has a very slight dependence on the minimum allocation, and in fact the delay curves are basically flat, whatever the number of levels, with maximum variations of about 5%.

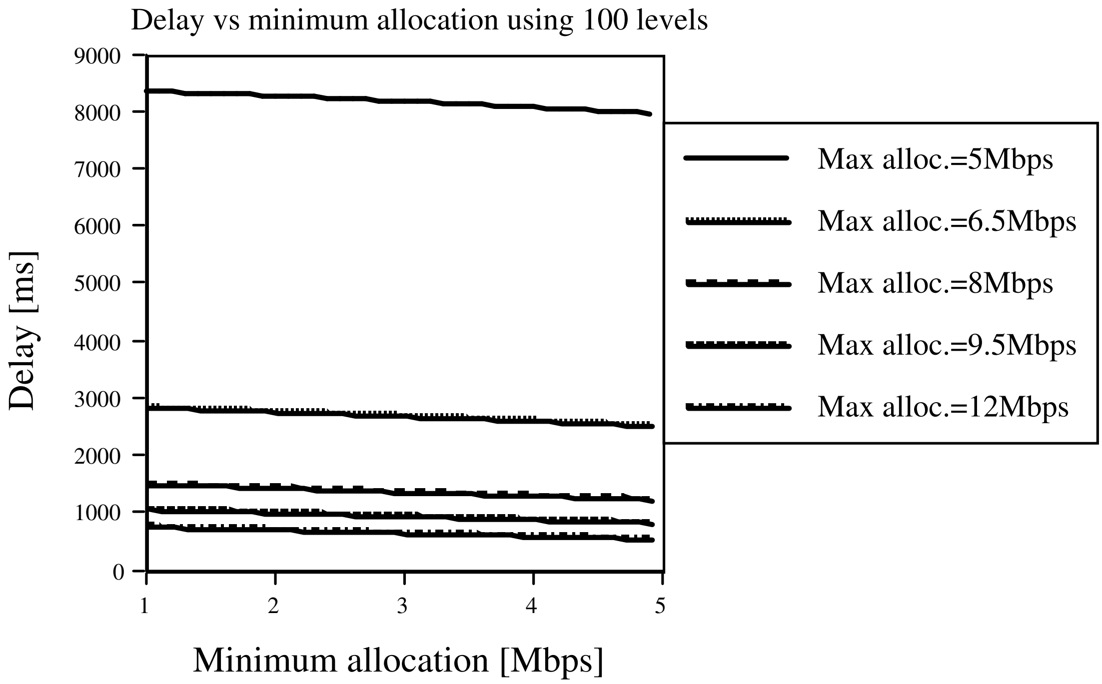
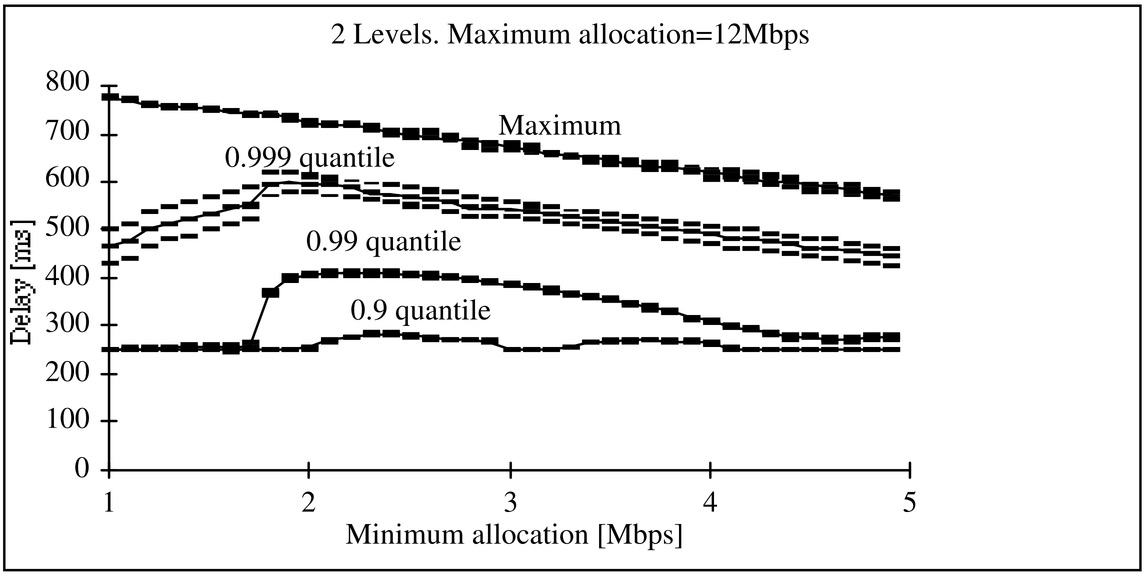
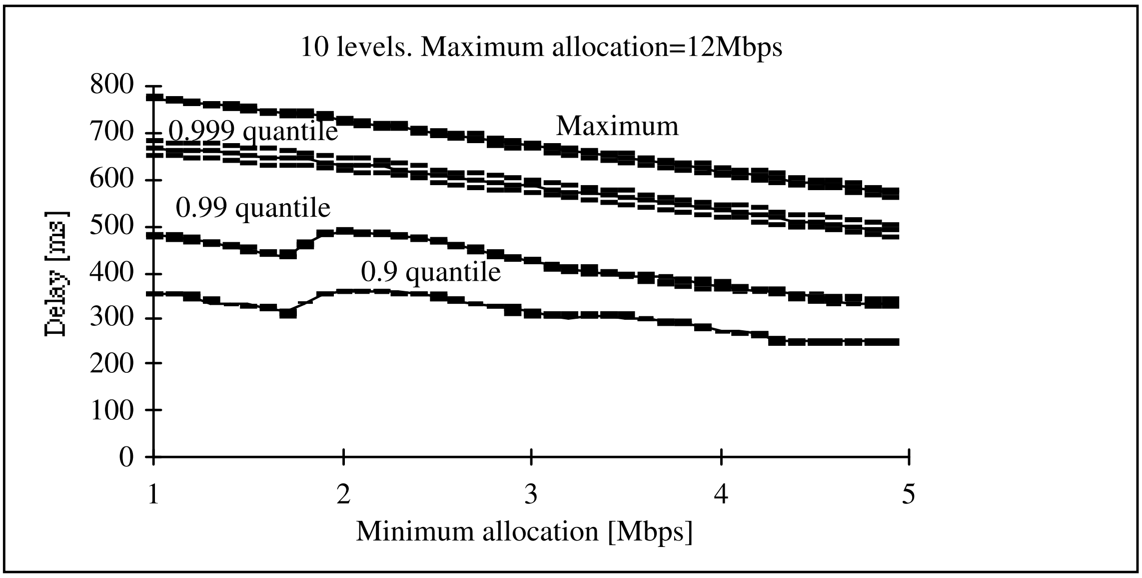



*Figure 4. Data delay vs. minimum allocation: 100 allocation levels with maximum allocations equal to 5, 6.5, 8, 9.5, and 12 Mbps.*

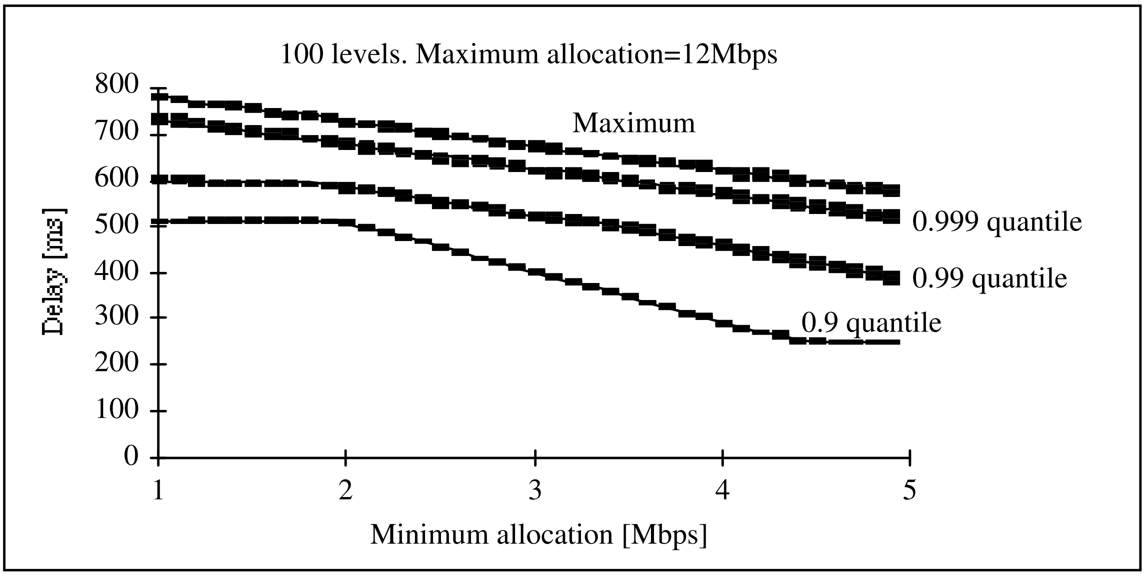
Let us consider now the case when the booked allocation *Amax* is greater than the peak VBR traffic throughput. In this case, the delay is influenced by the level switch delay alone, while in the cases presented above the delay is mainly influenced by *Amax*, and only secondarily by the level switches.


2 Levels. Maximum allocation=12Mbps
Delay [ms]
Minimum allocation [Mbps]
Maximum
0.999 quantile
0.99 quantile
0.9 quantile
800
700
600
500
400
300
200
100
0
1
2
3
4
5

*(a)*


10 levels. Maximum allocation=12Mbps
Delay [ms]
Minimum allocation [Mbps]
Maximum
0.999 quantile
0.99 quantile
0.9 quantile
800
700
600
500
400
300
200
100
0
1
2
3
4
5

*(b)*




*(c)*

*Figures 5a-c. Data delay vs. minimum allocation with 2, 10, 100 allocation levels. The continuous line connects the mean values obtained with simulation. 95% confidence intervals are shown.*

A characteristic common to the plots of the delays obtained by varying the minimum allocation is the presence of a slight raising in the central part, which is highlighted in Figures 5a-c.

This effect is most clearly visible with a small number of allocation levels, and is due to a greater frequency of level switches, because of the characteristics of the input traffic. Looking at the figures that show the delay vs. the minimum allocation, and, more clearly (as different scales are used) in the figures with the maximum allocation equal to 12 Mbps, we can see that the raising point corresponds to *Amin* values such that the peaks of the input distribution do not fall entirely between two allocation levels. This effect is justified by the fact that the short-term autocorrelation of the output states of the Markov chain is quite high. In fact, when in the second state, whose steady state probability is 0.64, the probability of staying in the same state in the next GOP is 0.93; when in the third state, whose steady state probability is 0.34, the probability of staying is 0.85. Thus, for a low number of levels and a low *Amin* (lower than 1.7 Mbps), the allocation requested almost always corresponds to one of the highest levels, thus decreasing the number of level transitions and consequently the end-to-end delay. When the minimum allocation increases, so does the probability of transitions, and their influence on the delay. Such an influence again decreases for a greater *Amin*. Notably, this effect tends to disappear when the number of levels is high.

The lines describing the behaviour of the maximum delay do not show the same risings, and decrease as *Amin* increases. This is due to a situation that only occurs at the beginning of the movie. In fact, the implementation of the VnL-DA algorithm assigns *Amin* as the first allocation, while the movie (and thus our model) starts with a throughput that is very close to the maximum. Therefore, at the beginning of the simulation there is always a very high delay, which is inversely proportional to the minimum allocation *Amin*. When *Amin* is equal to the peak throughput of the input traffic, the maximum delay is extinguished, in the sense that it becomes equal to the round trip time.

The study of the packet delay with the maximum booked bandwidth is above all useful for assessing the correctness of the simulation, and to set a limit on the effect of the delay induced by the level switching. The conclusion is that neither the number of levels *nlev* nor *Amin* play a significant role as far as the packet delay is concerned. Moreover, setting the booked bandwidth *Amax* to values greater than 8 Mbps does not produce any appreciable improvement.

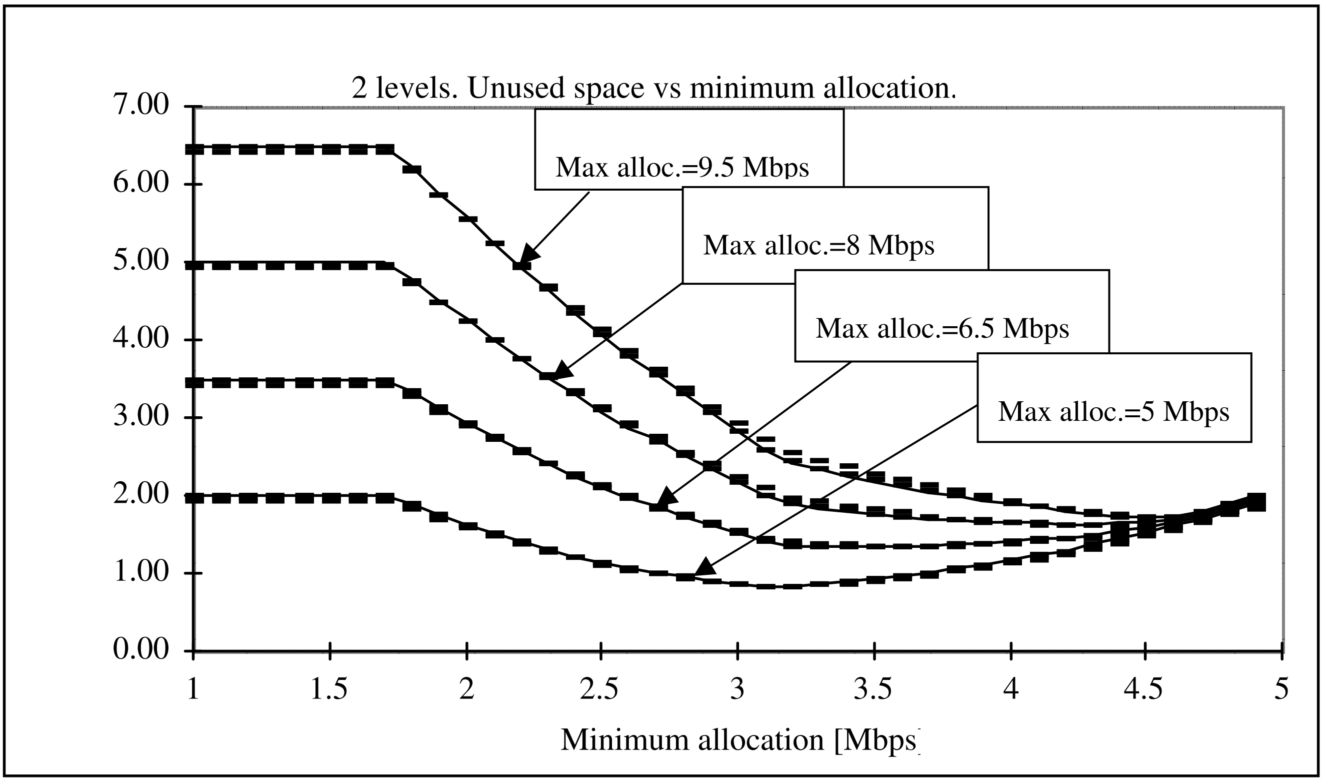
### *6.2 Unused allocation space*

It is possible to estimate the unused space by using the statistical properties of the synthetic input traffic. In fact, the unused space *u* is well approximated by the difference between the input traffic and the relative allocation request, weighted with the probability of a given input traffic value. Denoting by *t* the input throughput, we have:

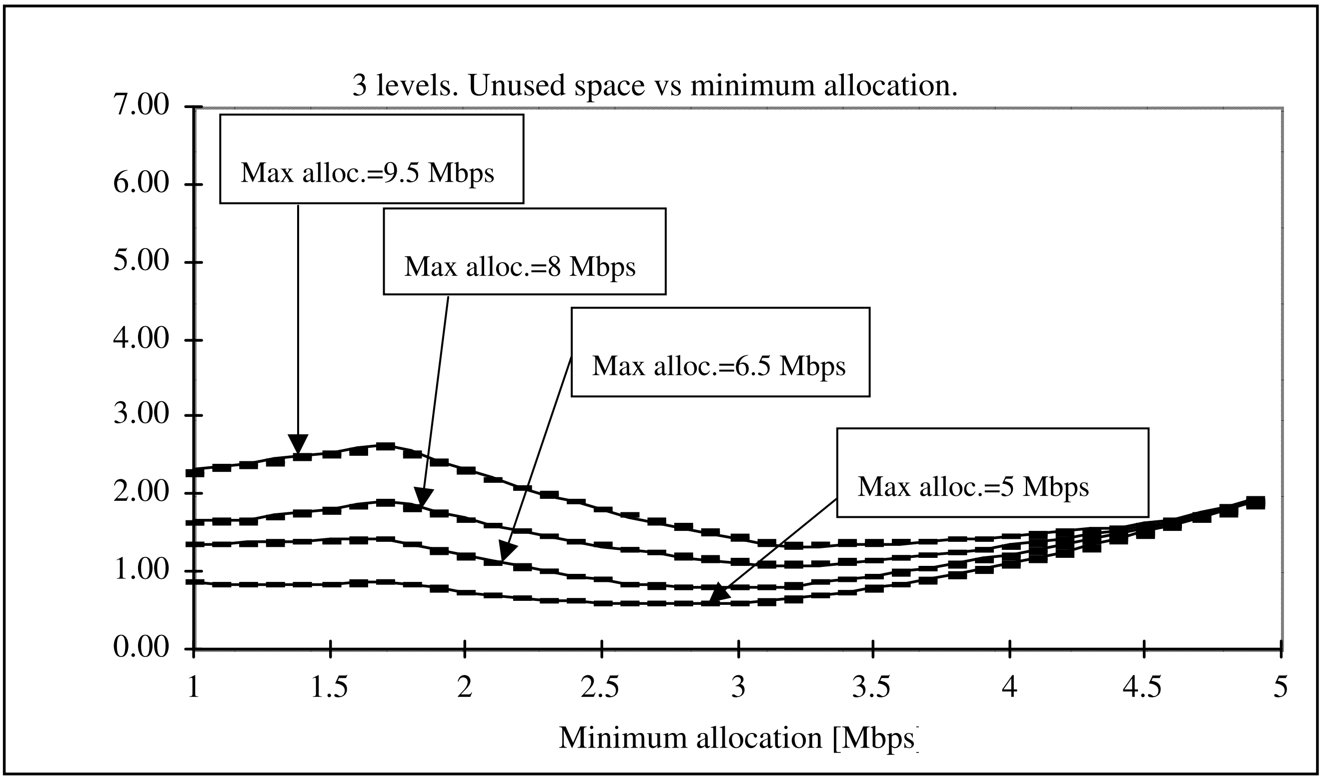
$$u = \int_{min(t)}^{max(t)} [A(t) - t] P(t),$$

where $A(t)$ is the allocation relative to a given value of the input throughput.

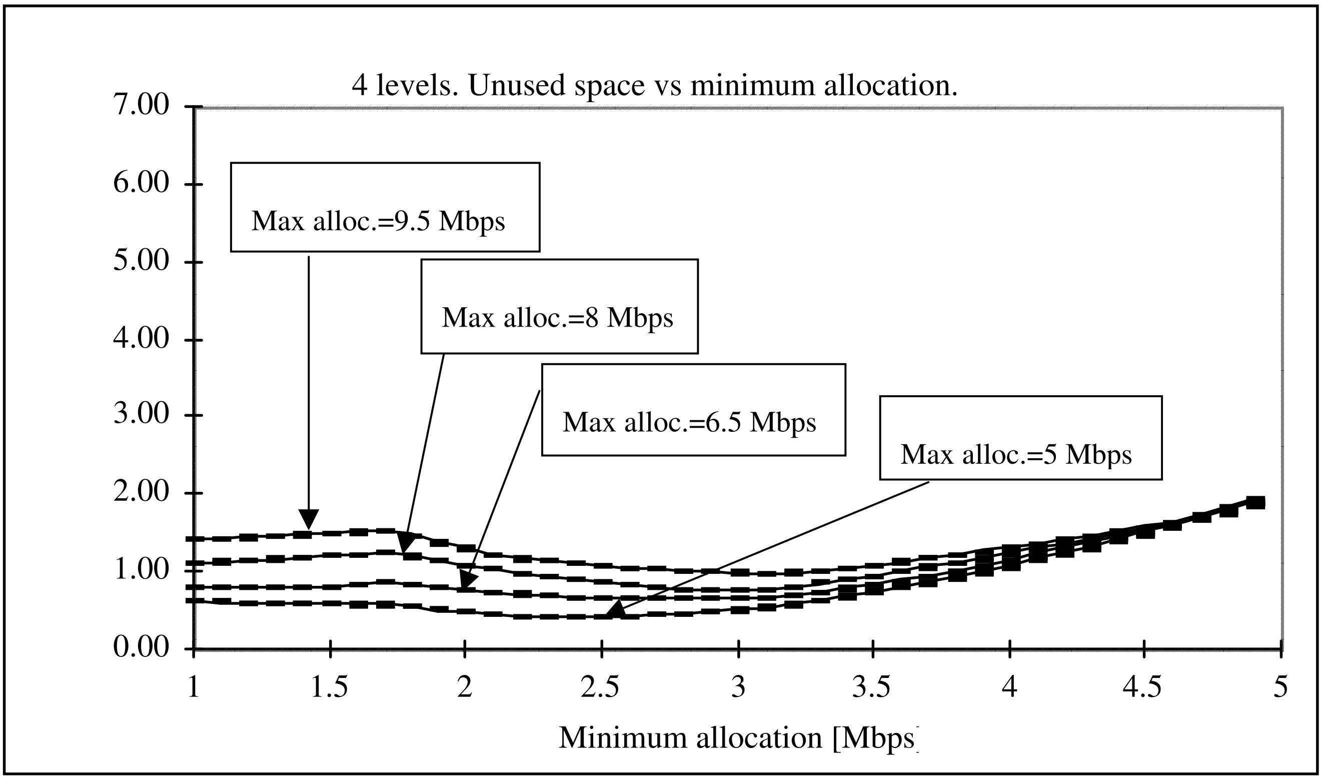
The unused allocation space is plotted in Figures 6a-e for various *Amax* values, as a function of *Amin*. Computed values are depicted as solid lines, while the simulation results are reported as the upper and lower bounds of 95% confidence intervals. The matching is excellent, in spite of the analytical model being only an approximation of the allocation algorithm. While the number of levels is low, the unused space very much depends on the booked allocation, which is lower for low *Amax*. However, the above considerations on the packet delay mean that the cases where *Amax* is less than 6.5 Mbps should be discarded.


2 levels. Unused space vs minimum allocation.
Max alloc.=9.5 Mbps
Max alloc.=8 Mbps
Max alloc.=6.5 Mbps
Max alloc.=5 Mbps
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
1
1.5
2
2.5
3
3.5
4
4.5
5
Minimum allocation [Mbps]

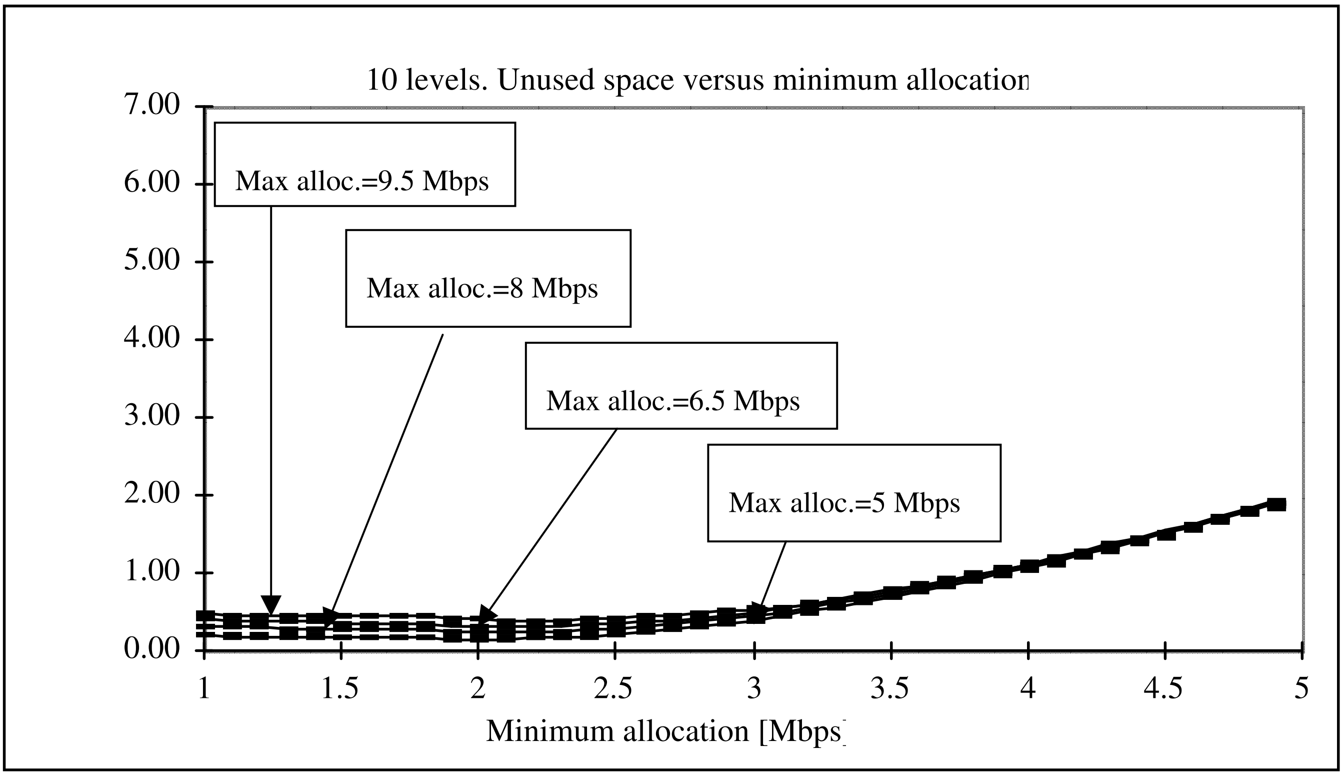
*(a)*


3 levels. Unused space vs minimum allocation.
Max alloc.=9.5 Mbps
Max alloc.=8 Mbps
Max alloc.=6.5 Mbps
Max alloc.=5 Mbps
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
1
1.5
2
2.5
3
3.5
4
4.5
5
Minimum allocation [Mbps]

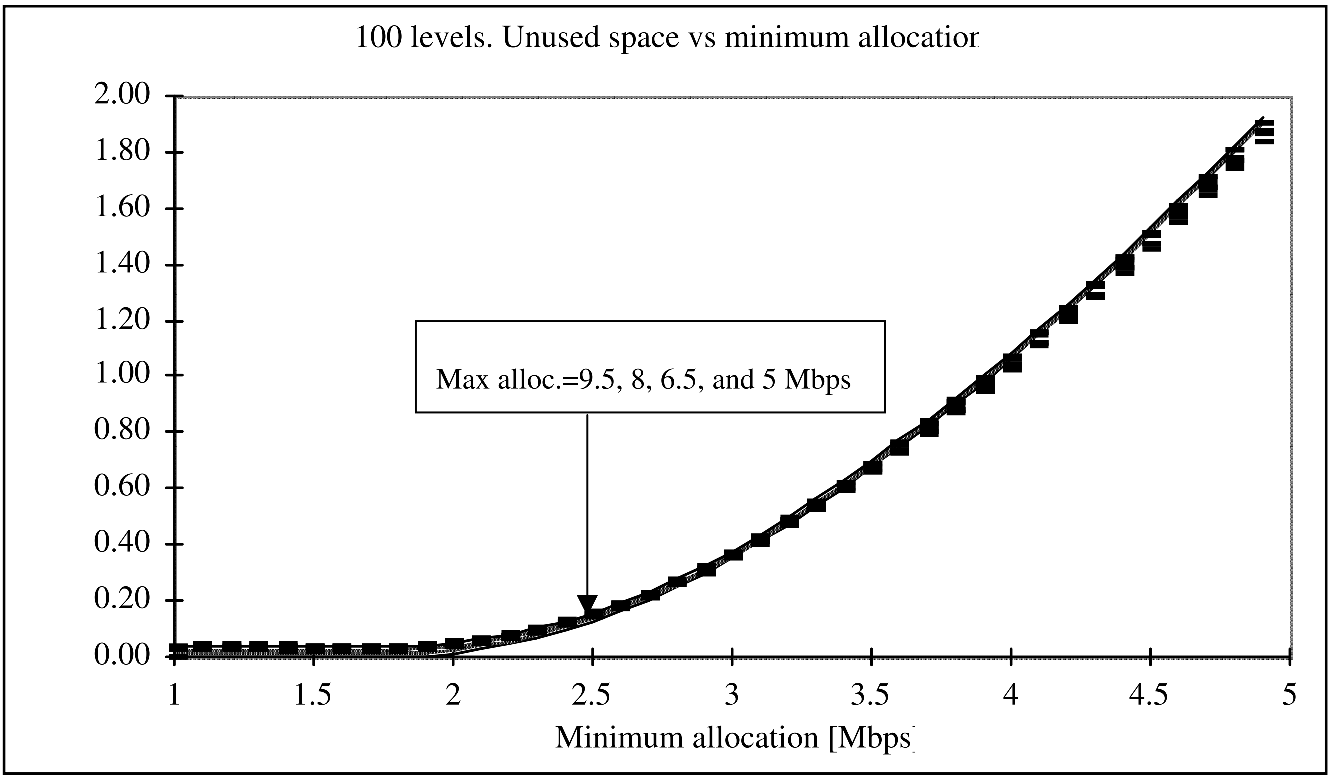
*(b)*


4 levels. Unused space vs minimum allocation.
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
Max alloc.=9.5 Mbps
Max alloc.=8 Mbps
Max alloc.=6.5 Mbps
Max alloc.=5 Mbps
1
1.5
2
2.5
3
3.5
4
4.5
5
Minimum allocation [Mbps]

*(c)*


10 levels. Unused space versus minimum allocation
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
Max alloc.=9.5 Mbps
Max alloc.=8 Mbps
Max alloc.=6.5 Mbps
Max alloc.=5 Mbps
1
1.5
2
2.5
3
3.5
4
4.5
5
Minimum allocation [Mbps]

*(d)*




*(e)*

*Figures 6a-e. Two, three, four, ten, and hundred levels. Unused space vs. minimum allocation for various values of the maximum allocation with 95% confidence intervals is shown with dashes. The continuous lines are computed values.*

For the case of *nlev* = 2, the unused space very much depends on the minimum allocation. In fact, by choosing the right *Amin* value, we obtain values of unused space which differ by more than 50%, for a given *Amax*. The two most significant cases, as far as both the unused space and the delays are concerned, are obtained with *Amax* set to 6.5 and 8 Mbps, while the cases with *Amax* set to 5 and 9.5 Mbps are less efficient, either in terms of packet delay or unused space. In both the interesting cases for *nlev* = 2, the minimum wasted allocation corresponds to *Amin* set to 3.3 Mbps, which is higher than the mean throughput of the input traffic. The minimum is present where there are 2 and 3 levels, while the curves flatten with increasing numbers of levels. Since the point of minimum is strongly dependent on the input distribution, it is preferable to have flat curves, because it makes the system performance less dependent on the particular MPEG model used to tune it. Therefore, the major gain of using many levels is not so much in the improvement of the optimal point but in the greater flatness of the lines that describe the unused space. At higher numbers of levels the dependence on *Amax* tends to disappear, and the efficiency of the algorithm improves as *Amin* is smaller. This also means that the dependence on the characteristics of the input generator is practically lost. For 100 levels, which is an approximation of an infinite number of levels, the unused space tends to zero for low *Amin*, as was expected.

## 7. Conclusions

The simulation study shows that neither the number of levels *nlev* nor *Amin* play a significant role as far as the packet delay is concerned. The major gain of using many levels is in the diminished dependency of the unused space on the minimum allocation Amin. At higher numbers of levels the dependence on *Amax* tends to disappear, and the efficiency of the algorithm improves as *Amin* is smaller. This means that the dependence on the characteristics of the input generator is practically lost because a higher number of levels means that the system adapts itself closely to the behaviour of the source, so the efficiency is independent of it. This is an interesting result, given the widespread interest in characterizing the behaviour of the VBR video sources.

For 100 levels, which is an approximation of an infinite number of levels, the unused space tends to zero for low *Amin*, as was expected. The channel efficiency is very high, with very little unused allocation space when many allocation levels are used, and virtually no allocation waste when the number of levels is in the order of 100.

When using a centralized allocation scheme with an allocation delay of 500 ms, we managed to keep the queuing delay under 500 ms 99.9% of the time by booking a channel share of two to three times the mean rate of the VBR flow, and 100% of the time by booking four times the mean rate of the VBR flow, i.e., the flow' peak rate. Whether it is necessary to book a bandwidth equal to the peak throughput of the VBR flow, or whether a smaller one is sufficient, depends on the ability of the receivers to cope with a given video frame loss rate.

We expected to find an optimum number of allocation levels, because we thought that the queuing delays introduced by allocation level switching would have made it impractical to use a high number of levels. Indeed, queuing delays with hundred levels are higher than delays with ten levels, but the relative difference is so small that it is practically negligible. In short, it is convenient to use as many allocation levels as possible, at least up to a value of around 100. If using a high number of allocation levels is impossible because of link layer limits on the minimum allocable unit, a minimum number of four levels should be used, which provides better performance and less dependence on the input pattern than the V2L method proposed in [14].

## References

[1] ATM forum, "Traffic management specification", version 4.0 1996, see also "ATM service categories", ATM forum EMAC white papers 1996.

[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, ”RSVP: a New Resource ReSerVation Protocol”, IEEE Network, pp. 8-18. September 1993.

[3] H. Zhang, ” Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks”, Proceedings of the IEEE, Vol. 83, No. 10, pp. 1374-1396, October 1995.

[4] F. Reichmeyer, L. Ong, A. Terzis, L. Zhang and R. Yavatkar, "A Two-Tier Resource Management Model for Differentiated Services Networks", IETF internet draft (work in progress), November 1998.

[5] N. Celandroni, E. Ferro, F. Potortì: “FEEDERS-TDMA: A distributed-control algorithm for satellite channel capacity assignment in a mixed traffic and faded environment”, International Journal of Satellite Communications, Vol. 15, No. 4, pp. 185-195, July-August 1997.

[6] N. Celandroni, E. Ferro, F. Potortì: “DRIFS-TDMA: A proposal for satellite access distributed-control algorithm for multimedia traffic in a faded environment”, International Journal of Satellite Communications, Vol. 15, No. 5, pp. 227-235, September-October 1997.

[7] N. Celandroni, E. Ferro, N. James, F. Potortì: “FODA/IBEA: a flexible fade countermeasure system in user oriented networks”, International Journal of Satellite Communications, Vol. 10, No. 6, pp. 309-323, November-December 1992.

[8] T. Zein, G. Maral, T. Brefort, M. Tondriaux: “Performance of the Combined/Fixed Reservation Assignment (CFRA) scheme for aggregated traffic”, Proceedings of the COST-226 Final Symposium, pp.183-198, Budapest (H), 10-12 May 1995.

[9] Dornier Deutche Aerospace: “Advanced Business Communications via satellite”, System Description, July 1992.

[10] Jahangir I. Mohammed, Tho Le-Noc: “Performance analysis of Combined Free/Demand Assignment Multiple Access (CFDAMA) Protocol for Packet Satellite”, ICC 1994, pp. 869-873.

[11] Tho Le-Gnoc, S.V. Krishnamurthy: “Performance of Combined Free/Demand Assignment Multi-Access (CFDAMA) Protocol with Pre-Assigned Request Slots in Integrated Voice/Data Satellite Communications”, Proceedings of the International Conference on Communications, ICC'95, pp. 1572-1576, 1995.

[12] H.O. Awandala, L.G. Cuthbert, J.A. Shormans: “Improving the delay of best effort traffic over ATM satellite networks”, Electronics Letters, No. 17, Vol. 35, pp. 1430-1432, 1999.

[13] N. Celandroni, E. Ferro, F. Potortì, A. Chimienti, M. Lucenteforte, R. Picco: “MPEG-2 coded video traces transmitted over a satellite link: scalable and non-scalable solutions in rain fading conditions”, Multimedia Tools And Applications, Vol. 10, N. 1, pp. 73-97, 2000.

[14] N. Celandroni, M. Conti, E. Ferro, E. Gregori, F. Potortì: “Bandwidth assignment on a satellite channel for VBR traffic”, International Journal of Satellite Communications, Vol. 15, 1997, pp.237-246.

[15] A. Chimienti, M. Conti, E. Gregori, M. Lucenteforte, R. Picco, “MPEG 2 Variable Bit rate coding algorithm: analysis and modeling”, SPIE proceedings, Vol. 2952, N. Otha (editor), pp.597-607.

[16] M. Conti, E. Gregori, “Traffic characterization and bandwidth allocation for MPEG 2 VBR Video Traffic” Proc. GLOBECOM'97, Phoenix, Arizona, November, 1997, pp. 443-447.

[17] M. Conti, E. Gregori, A. Larsson: “A study of the impact of MPEG-1 correlations on video-sources statistical multiplexing”, IEEE Journal on Selected Areas in Communications, Vol. 14, September 1996, pp.1455-1471.

[18] N. Celandroni, E. Ferro, F. Potortì: “A simulation tool to validate and compare satellite TDMA access schemes", Telecommunication Systems, 12 (1999) 1, pp. 21-29.

[19] H.O. Awandala, L.G. Cuthbert, J.A. Shormans: “Framework for supporting real-time VBR traffic over satellite networks”, Electronics Letters, No. 8, Vol. 35, pp. 616-617, 1999.