Aggregation of Markovian Sources: Approximations with Error Control

Marco Conti, Silvia Ghezzi, and Enrico Gregori

CNUCE, Institute of National Research Council
Via S. Maria 36 - 56126 Pisa - Italy
{Marco.Conti, Enrico.Gregori}@cnuce.cnr.it

Abstract. One of the most challenging applications for high speed broadband networks is VBR video-conference and movies. Markovian models are a flexible tool to characterize a wide set of multimedia sources, including VBR video traffic. Specifically, these models are efficiently used for the description of a single video source. Nevertheless, teletraffic studies with superposed markovian sources have a computational complexity that grows exponentially with the increase of the number of the sources and this reduces the usefulness of these models in such an environment. The problem of characterizing the traffic generated by *N* superposed independent sources is addressed in the paper. Specifically, in the paper we define an aggregation method for the description of this kind of traffic. The state space of our aggregate model strictly depends on the required level of accuracy. Furthermore, we can decide a-priori the complexity of the aggregate model as function of the precision we want the model to have.

1 Introduction

One of the most interesting and emerging application is Variable Bit-Rate (VBR) video. This application is characterized by both a complex traffic profile and stringent QoS requirements (low packet-loss rate and low transmission delays). To efficiently provide the QoS required by these kind of applications, a very accurate video traffic model is needed. For this reason the modeling of VBR video sources has recently received significant attention. Since MPEG coding algorithms are becoming the standards for VBR video coding [2], in this paper we will model VBR video sources as MPEG sources. The model has been proposed, utilized and validated for describing VBR video sources, coded by an MPEG coder.

At the present, there is currently not a widely accepted model and three types of models have been proposed for VBR sources: autoregressive models ([12], [14]), Markov chain based models ([4], [13]) and models which capture properties of self-similarity and long-range dependence [7]. Although Markov chains do not have long-

Work carried out under the financial support of CNR, in the framework of the project “Sistemi radiomobili multimediali nell’evoluzione verso UMTS”.

G. Pujolle et al. (Eds.): NETWORKING 2000, LNCS 1815, pp. 350-361, 2000

range dependence as self-similar models, in [10] it is shown that VBR video sources are adequately modeled by a Markov chain as long-range dependence is not a crucial property in the teletraffic studies of interest. Furthermore, Markov models have been shown to provide an adequate traffic characterization for statistical multiplexing studies. Therefore in this paper we assume the use of Markov chains to predict the statistical properties of VBR video sources.

In this work, we start from a markovian characterization of a single source. Then we present and validate a method for the construction of a Markov chain describing the traffic generated by a set of superposed *i.i.d.* sources. We first show an *exact model* for the description of the traffic generated by $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ superposed sources, which consists of the simple overlap of *N* single-source traffic models. In this paper we propose an approximate *aggregate m*odel to characterize the traffic generated by N superposed sources..Specifically, we develop, by using the single-source model, a multiple-source model based on the aggregation of the output traffic of *N* superposed video sources. The aggregate model accomplishes a balance between the complexity and the accuracy level the model will have. Later on, we give a comparison between this aggregate approach and the exact one by measuring the basic statistics such as mean bit rate, standard deviation, distribution and autocorrelation function.

2 Markovian Modeling of VBR Video Sources

The aim of this work is to find a suitable characterization of the video traffic generated by *N* independent and identical sources. Analog video signals are digitized and compressed before being transmitted over communication networks. Current compression techniques usually yield VBR video traffic. In this model the process $\{H_k | k \geq 0\}$ describes the total amount of bits generated by a VBR source in a time unit. To avoid unnecessary complexity (in the state space of $\{H_k | k \geq 0\}$), the bit rate information is quantized into a number *M* of levels, and the quantization levels are defined in a uniform way: the possible bit rate per time unit H_k is included in a range delimited by a minimum value l and a maximum value u; the quantization levels are obtained by dividing the range $u - l$ per M. The transition probabilities $P\{H_k | H_{k-1}\}$ are computed through a movie trace and occur every time unit. The structure of the resulting Markov transition matrix is tridiagonal, or block tridiagonal [1], and this structure attests the clustering of bit rates within a scene, while the number of states in the model indicates the range of scene types. Furthermore, transition probabilities are concentrated on the diagonal (0.95) that indicates a long sojourn time in a single state and a high value of correlation, which are typical characteristics of markovian chains in the modeling of VBR video sources [4]..The steady state probabilities for the bit rate H, with a value of *M* equal to 7, are shown in Figure 1.

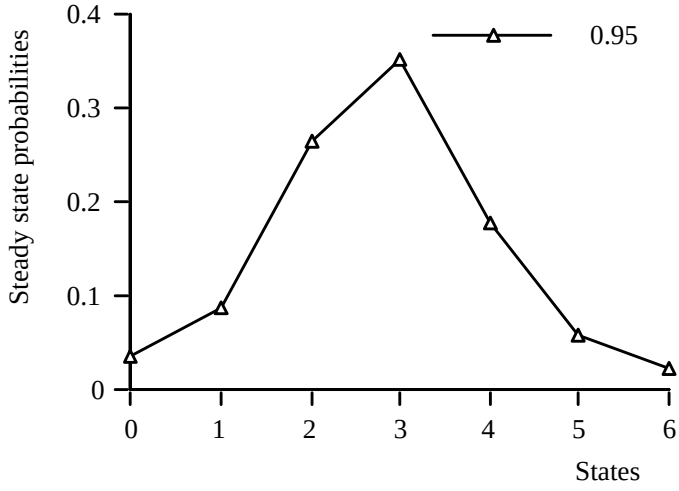


Fig. 1: Density of the steady state probability

This is a general model and one of its most practical utilization is the characterization of sources coded according to an MPEG encoder. An MPEG encoder produces a pattern that is periodically repeated. This pattern is referred to as GOP (group of picture) and it often made up of 12 frames. Previous studies [4] show that this single-source model is able to capture the properties of a VBR source such as steady-state probabilities and basic statistics. The auto-correlation of the GOP sequences generated by the model has a very slow decay. In the following, we use this single-source markovian model to design a model for the characterization of N superposed *i.i.d.* sources.

2.1 Exact Model

We recall that the system we aim to describe consists of N *i.i.d.* sources, that is we have N traffic flows, each of them generated according to the same distribution and in an independent way. The straightforward way to describe the N sources is to compute a transition probability matrix obtained from the Kronecker product of the N single-source Markov chains. The final matrix should contain the union of all possible states in which the N sources can pass.

Let N be the number of sources, each of them is described by its proper bit rate per time unit H_k $(k = 1,\ldots,N)$. For a N-source system, if we choose to quantize the bit rate H_k in M levels then the state space of the resulting Markov chain consists of M^N possible states. In this paper to produce numerical results we assume $M = 7$.

The exact model describes the N video-sources system by representing the characteristics of each source in isolation, but the state space of the Markov chain increases exponentially with the number of sources, because of the Kronecker product.

The dimension of the transition probability matrix ($7^N \times 7^N$) makes this approach useless when the number of sources grows. When we have i.i.d. sources the complexity of the aggregate source is lower then that obtained via the Kronecker product as state which are simple permutation can be merged together. In this paper we propose an alternative characterization method that further reduces the state complexity. In the following, we refer to the Kronecker-product model as *exact model* to distinguish it from the model we propose to solve the complexity problem. Our methodology is based on the definition of an aggregate model as expressed in the next section.

2.2 Aggregate Model

In the existing literature, several methods have been proposed in order to reduce the complexity of stochastic models. Among them, we focus on *aggregation techniques* where the original system model is replaced with a reduced model which mimics the behaviour of the original one [5]. According to this method we aim to characterize the system by having a single stochastic aggregation process able to exhibit the same behaviour as several *i.i.d.* stochastic processes. The aggregate process is *flow equivalent* to the N superposed *i.i.d.* processes [11], that is at any time unit the traffic arrival rate of the aggregate process is the same than the one of the N processes together, but it requires less information with respect to the exact one.

Starting from the real system, whose space state consists of all the possible N-tuples $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$, we aim to design an aggregate model whose states are obtained by grouping together N-tuples that are equivalent from both the total bit rate and their future evolution standpoint. Specifically, if two N-tuples $\overline{H}_i = \left[H_1^{(i)}, H_2^{(i)}, \ldots, H_N^{(i)}\right]$ and $\overline{H}_j = \left[H_1^{(j)}, H_2^{(j)}, \ldots, H_N^{(j)}\right]$ represent two states of the real system where the N sources *i*) generate the same total bit rate per time unit and *ii*) follow the same probabilistic evolution, then we want to collect these N-tuples in the same state of the aggregate model. Thus, while the state space of the exact model consists of $7^N \times 7^N$ states, the aggregate model will have a reduced number of states and the dimension of the aggregate model state space depends on the level of approximation and the computational cost we want to have.

To summarise, the basic idea of the aggregate model is to group original states together taking into account the total bit rate and the evolution of the N-tuples in system. However, while it is easy to understand how to estimate the total bit rate corresponding to an N-tuple, it is not straightforward to define the rule which assesses the evolution of an N-tuple in the system.

2.3 Maintaining the Autocorrelation of the System

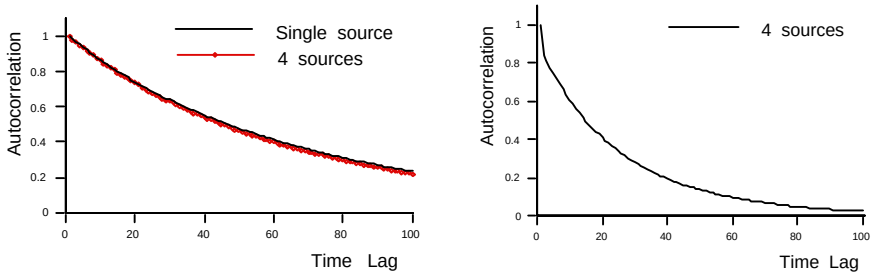
To better explain why it is important to group together states with the same evolution in the system, let us see an example.

Let us assume to group together in a single aggregate state the states $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ of the real system characterized by the same values of $\overline{D_q}$, where $\overline{D_q}$ is the quantized value of $\overline{D}_k$ and.

$$\overline{D_k} = \frac{\sum_{j=1}^{N} H_j^{(k)}}{N} \quad ;$$

for each state $\overline{H}_k$ in the aggregate state characterized by $\overline{D_q}$.

It is evident that all the states $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ with the same value of $\overline{D_q}$, are very similar from the traffic flow standpoint because they generate approximately the same total bit-rate at each time-unit (the approximation is due to the quantization of $\overline{D}_k$). Anyway, some of the N-tuples $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ belonging to an aggregate state can be different as regards their evolution in the system.



part a) **part b)**

Fig. 2. Autocorrelation in the Exact Model (part a) vs Aggregate Model (part b)

We want to show now that if N-tuples $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ are collected in an aggregate state only according to their total bit rate, then, when evolving, the aggregate state can generate a trajectory that is not compatible with the real system and that does not maintain its autocorrelation . For example, let a system be composed by 4 sources and let $\overline{D}_i$ be quantized in 7 levels and suppose to adopt the uniform quantization strategy, as for H_k. Then, according to the aggregation and to the uniform quantization processes, the N-tuples [6600] and [3333] collapse into the aggregate state corresponding to $\overline{D}_q = 3$. These two N-tuples, corresponding to two states of the real system, are equivalent as far as the traffic flow, because they generate the same amount of traffic, but they are different as far as their trajectories. In fact, the sources in the N-tuple [3333] will probably pass in different states at the following transition step, by comparison with what will happen to the sources in the N-tuple

[6600]. This misconduct of the aggregate model can bring the model to a loss of autocorrelation because we merge states which should have different evolution to the same aggregate state, and this aggregation leads not to maintain the real correlation of the original system. Conversely, if we study the autocorrelation of the aggregate system when grouping together states like the two N-tuples [6600] and [3333], which are similar from the bit rate standpoint but different from the evolution standpoint, we have a very evident loss of correlation, see Figure 2 (part b). Thus, we can assert that this aggregate model is not able to maintain the autocorrelation in the system.

2.4 Distance Between States

In order to keep the autocorrelation as close as possible to the one of the real system, we want to group together only states that are equivalent from both the aspects we analyzed: traffic, i.e. rate per time unit, and autocorrelation.

The ideal situation is when an aggregate state only contains states which generate exactly the same total bit-rate per time unit and which evolve following the same trajectories at each transition step.

What we need first is a method able to estimate the *similarity* of two states in term of their evolution. After that, we only need to group together *similar* states. The basic idea is to identify a parameter able to capture this states' similarity. Since $\overline{D_q}$ well describes the traffic behaviour we only add a new coefficient, called ε, which estimates the level of "precision" in term of ability of the model in capturing the similarity among states. An aggregate state in the new model is thus designated by the variable $\{\overline{D}_q\}_\varepsilon$. In order to give a better explanation of this concept we will give some definitions.

In the following, we will use "N-tuple" at the same place as "state", and with "similar" we mean the similarity among states in term of similar trajectory in the system evolution.

Definition 1. A state $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ is called a permutation of another state $\overline{H}_j = \left[H_1^{(j)}, H_2^{(j)}, \ldots, H_N^{(j)}\right]$ when the values $H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}$ and $H_1^{(j)}, H_2^{(j)}, \ldots, H_N^{(j)}$ are exactly the same and only differ for the order in the two N-tuples respectively. For example, the N-tuple [1262] is a permutation of [6221].

Definition 2. An N-tuple $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ is called to be in a lexicographic order if the values $H_i^{(k)}$ are not increasing when the value of i increases.

For example, the lexicographic order of the N-tuple [1262] is [6221].

Definition 3. Given two N-tuples $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ and $\overline{H}_j = \left[H_1^{(j)}, H_2^{(j)}, \ldots, H_N^{(j)}\right]$ in lexicographic order, then the *distance* $d(\overline{H}_k, \overline{H}_j)$ between the two N-tuples is defined as: $d(\overline{H}_k, \overline{H}_j) = \sum_{i=1}^{N} \left| H_i^{(k)} - H_i^{(j)} \right|$.

For example, the distance between the two N-tuple [6221] and [4440] is equal to 7.

Lemma 1 *The distance between two N-tuples $\overline{H}_k$ and $\overline{H}_j$ which are permutation between each other is $d(\overline{H}_k, \overline{H}_j) = 0$.*

For example, the distance between the two N-tuple [6221] and [1262] is equal to 0.

The demonstration of this Lemma is straightforward if we first rearrange the two N-tuples in a lexicographic order.

Lemma 2 *Two states $\overline{H}_k$ and $\overline{H}_j$ whose distance is $d(\overline{H}_k, \overline{H}_j) = 0$ are* equivalent *from the evolution in the system standpoint.*

The demonstration of this lemma is straightforward because all of the *N* sources are *i.i.d.* .

It's worth noting that there is no difference both in term of generated traffic and in term of their probabilistic evolution between two *N*-tuples which are permutation one each other.

Definition 4. Two states $\overline{H}_k$ and $\overline{H}_j$ are defined *similar* when their distance is $d(\overline{H}_k, \overline{H}_j) = 0$

It is now possible to define the key idea that underlines our aggregation methodology. The methodology aims to define each aggregate state I by collecting states $\overline{H}_j$ with equivalent bit rate per time unit and with a distance $d(\overline{H}_k, \overline{H}_j)$, with any other $\overline{H}_k$ in I, as low as possible. If this happens, we maintain the autocorrelation of the system because we group together states which are very similar from the evolution standpoint.

Definition 5 Let I be an aggregate state, which groups together N-tuples $\overline{H}_k$. We define ε as an estimation of the maximum possible distance $d(\overline{H}_k, \overline{H}_j)$ between each pair of states in I:

$$\varepsilon = \max \left\{ d(\overline{H}_k, \overline{H}_j) \right\}, \ \forall \overline{H}_k, \overline{H}_j \in I \qquad \textbf{(1)}$$

ε varies in the range $[0, N \cdot (M-1)]$, where M is the number of quantization levels and N is the number of sources.

According to this definition we can design an aggregate model with a target level of precision, which is at least ε, and where a state is function of ε and $\overline{D}_q$. In Table 1 we show a tradeoff between the complexity of the state space of the aggregate model and the precision obtained by varying the value of ε and using the case study introduced in Section 2.3.2, i.e. 4 sources and 7 levels of quantization. The state space dimension for the exact model is 2401 states which reduces to 210 states by simply gouping permutations.

In the following we present the procedure that implements the aggregation mechanism, and then we propose an example of an aggregate model.

Procedure of the aggregation process, for a given value of ε:

1) For each possible state $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ calculate the quantized value $\overline{D}_q$;

2) By grouping together all states $\overline{H}_k$ with the same value of $\overline{D}_q$ create M subsets I of states;

For each subset I do:

3) rearrange each N-tuple $\overline{H}_k$ in lexicographic order;

4) for each $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ in lexicographic order, calculate the corresponding decimal representation $dec = \sum_{i=1}^{N} H_i \cdot 10^{N-i}$;

5) Order each *N*-tuple $\overline{H}_k$ in increasing order with respect to dec;

6) Aggregate according to the value of ε:

6.1) for each *N*-tuple in I calculate $d(\overline{H}_i, \overline{H}_{i+1})$ and do $d_s = \sum_i d(\overline{H}_i, \overline{H}_{i+1})$ while $d_s \le \varepsilon$;

6.2) Let j be the index by which $d_s = \sum_{i=x}^{j} d(\overline{H}_i, \overline{H}_{i+1}) > \varepsilon$, then group together all states from $\overline{H}_x$ to $\overline{H}_j$, $j = x$, and $d_s = 0$;

6.3) goto 6

Let M be the number of quantization levels of $\overline{D}_k$, the mean bit rate per time unit generated by a state $\overline{H}_k = \left[H_1^{(k)}, H_2^{(k)}, \ldots, H_N^{(k)}\right]$ (as defined in Section 2.3), and let N be the number of sources. Then the number of states of the real system is M^N. Furthermore, we choose to quantize the value of $\overline{D}_k$ in a uniform quantization.

By looking at the aggregation algorithm, it is easy to note that there are two different aggregation levels, see Figure 3. The first one is performed at the step 2 of the procedure where we create M subsets of the original state space, according to the quantized value of $\overline{D}_k$. Hence, we obtain M aggregate states containing N-tuples characterized by the same value of $\overline{D}_q$. Then, at the step 6, we perform the second level of the aggregation process. This level is worked out by keeping into

of the aggregation process. This level is worked out by keeping into consideration the precision and the complexity degree we want to obtain. Note that according to the computation of d_s there are not any pair of N-tuples in I with a distance bigger than d_s. Hence, we first establish a value of ε, i.e. the upper bound on the *precision* the model will offer, and then we perform a further aggregation: for each of the M subsets, we create a further number *x* of subsets. In all of these *x* subsets all of the N-tuples have a maximum distance d_s from any other N-tuple in the same subset equal to ε. It is worth noting that *x* could be different for each of the M subsets and in each of the *x* subsets the real maximum distance between N-tuples could be less than the target maximum value ε. Let *i* be the index that allows us to select one of the *x* subsets, then an aggregate state is represented by the pair $\{\overline{D_q}, i\}$, where $i = 0,..,x-1$ and $\overline{D_q} = 0,..,M-1$. In the following we refer to this model as ε*-based aggregate model*. Examples of use of this model can be found in [6].

Table 1. Complexity vs. precision in the aggregate model

Precision level ε	0	1	2	4	8	12	20
Number of aggregate states	210	136	106	66	43	32	20

Table 1 displays the state space dimension of the aggregate model for each level of precision ε. Of course the maximum degree of precision is reached when $\varepsilon = 0$, because we only group together permutations of states identical from the evolution and the traffic standpoint, but, in this case, we obviously get the highest number of states. Nevertheless, if we want to have a model with the lowest complexity (low number of states) we will aggregate states according to an high value of ε.

3 Autocorrelation Study in the Aggregate Model

To check if the aggregate model well captures the autocorrelation we plot the autocorrelation function for a system with 4 sources and 7 uniform quantization levels. In the Figure 4, we plot the autocorrelation function related to the aggregate model obtained by varying the value of ε. It easy to note that when $\varepsilon = 0$ the autocorrelation strictly overlaps the one obtained with the exact model. Also when $\varepsilon = 1$ the autocorrelation is still well captured. Then we show the autocorrelation obtained by reducing the precision level, i.e. with $\varepsilon = 7$, in order to see how the autocorrelation decreases when we introduce approximation in the model.

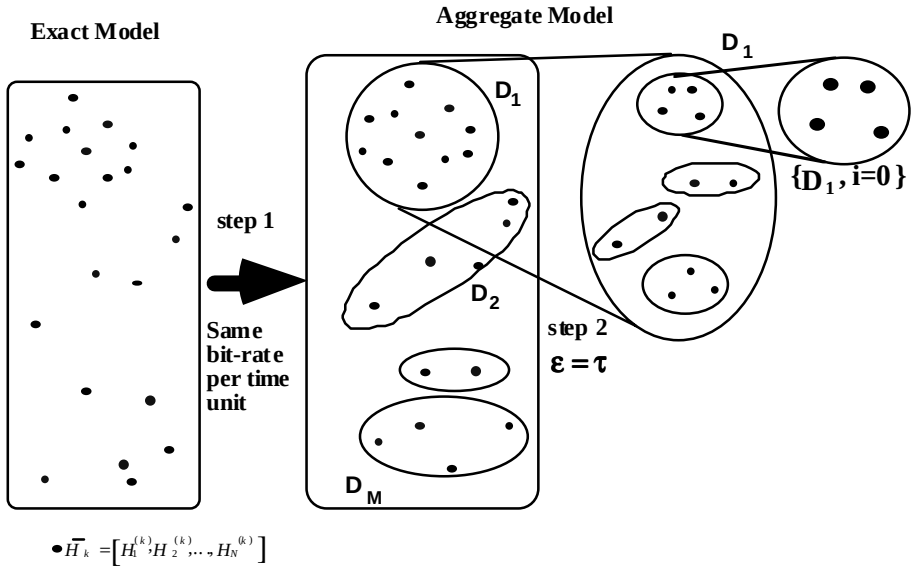


Fig. 3 Steps in the aggregation proces

We can see that, even in this case, the aggregate model is able to well capture the autocorrelation function: the loss of correlation is quite low comparing with the gain in term of space state complexity, see Table 2 for the number of states in the aggregate model. For all the other precision levels, $1 < \varepsilon < 7$, the autocorrelation is slowly getting worst as the value of ε increases, but always stays over the line plotted for $\varepsilon = 7$ .Furthermore, to check the efficiency of the aggregate model in capturing the statistical behaviour of the real system, we show in the Table 2 the basic statistics. Results are obtained from a 1.000.000-steps long trace generated by the exact model and by the aggregate model. The statistics *Mean*, *Variance*, and coefficient of variation c correspond to the four sources.

Table 2. Comparison between basic statistics in the Exact model and in the Aggregate model

	Exact Model	**Aggregate Model**		
		$\varepsilon = 0$	$\varepsilon = 1$	$\varepsilon = 7$
number of states	2401	210	136	46
Mean	11.26367	11.31	11.313	11.29
Var	5.94	6.2	6.1335	6.2
c	0.21636	0.22	0.219	0.22

Table 2 shows that the aggregate model is able to capture the basic statistics of the system independently from the complexity and the precision level we decide to have. That is, any value of ε we choose, the model we get well approximates the real system.

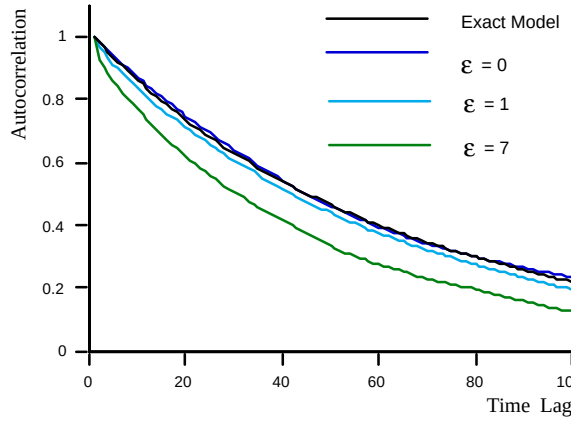


Fig. 4 Comparison between autocorrelation in the Exact model and in the Aggregate model

4 Conclusions

Modeling of superposed markovian sources is a difficult task due to state space complexity. In this paper, we have considered the modeling of N *i.i.d.* sources each characterized by the same Markov chain. We have showed a straightforward solution, *exact model*, consisting on the intersection of the N single Markov chains into one final Markov chain (obtained as the Kronecker product of the N single chains). The *exact model* is characterized by a state space which grows exponentially with the number of sources. To solve this problem we have proposed an alternative solution, *aggregate model*, whose key-point is the aggregation of the information of the original system, by providing a final Markov chain with a number of states lower than that of the exact model. Specifically, in the *aggregate model* we choose to describe the traffic flow generated by the system by using the information given by a pair of random variables $\overline{D}_q$ and ε which aims to aggregate together states similar from the amount of arrivals generated per time unit and their evolution in the system standpoint.

The complexity of this approach does not depend on the number of sources but only depends on the level of precision required. A comparison between the *exact model* and the *aggregate model* have been carried out. The analysis of the basic statistics shows that the aggregate model is able to capture the characteristics of the traffic

generated by the superposed sources, by providing a very good fitting as regards the distribution figures, mean, standard deviation and the autocorrelation function. Furthermore a bound aggregation model has been proposed which gives an overestimation on the traffic generated by the sources, and which is suitable to design bandwidth allocation algorithm based on an upper bound rate rather than on the peak rate. Results demonstrate that in some particular video sources configuration the gain we get with a bound allocation is quite good.

References

[1] K. Chandra, A.R. Reibman, "Modeling One- and Two-Layer Variable Bit Rate Video", *IEEE/ACM Transactions on Networking*, Vol. 7, no. 3, June. 1999, pp. 398-413.

[2] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG", IEEE Proceeding,Vol. 83, No. 2, February 1995, pp. 151-157.

[3] 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.

[4] M. Conti, E. Gregori, A. Larsson, "A study of the impact of MPEG-1 correlations on vodeo-sources statistical multiplexing", *IEEE Journal on Selected Areas in Communications*, Vol. 14, September 1996, pp.1455-1471.

[5] M. Conti, S. Ghezzi, E. Gregori, "A Methodology for an Efficient Characterization of Superposed Markovian Sources", *Proceedings of IFIP ATM'98 Workshop on Performance Modeling and Evaluation of ATM Netwroks*, July 20th-22nd, 1998.

[6] S. Ghezzi, "Methodologies for Efficient Characterization of Markovian Sources", Ph.D thesis (to be completed).

[7] J.J Gordon, "Long range correlation in multiplexed Pareto traffic", in *Proceedings of the International IFIP-IEEE Conference on Broadband Communications*, Canada, 1996

[8] D. Heyman, A.Tabatabai and T. Lakshaman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 2, No. 1, March. 1992, pp. 49-59.

[9] D. Heyman, T. Lakshaman, "Source model for VBR broadcast-video traffic", *IEEE/ACM Transactions on Networking*, Vol. 4, Feb. 1996, pp. 40-48.

[10] D. Heyman, T. Lakshaman, "What are the implication of long-range dependence for VBR video traffic engineering?", *IEEE/ACM Transactions on Networking*, Vol. 4, June. 1996, pp. 301-317.

[11] K. Kant, "Introduction to Computer System Performance Evaluation", *McGraw-Hill, Inc.*

[12] M. Nomura, T. Fujii and N. Ohta, "Basic Characteristics of Variable Rate Video Coding in ATM Environment", IEEE Journal on Selected Areas in Communications, Vol. 7, No. 5, June 1989, pp. 752-760.

[13] P. Pancha and M. El Zarki, "MPEG Coding For Variable Bit Rate Video Transmission", *IEEE Communications Magazine*, May 1994, pp. 54-66.

[14] G. Ramamurthy, B Sengupta, "Modeling and Analysis of a Variable Bit Rate Video Multiplexer", *Proceedings of IEEE INFOCOM*, Florence 1992, pp.817-827.

[15] O.Rose, "Simple and efficient models for variable bit rate MPEG video traffic", *Performance Evaluation*, vol. 30, pp. 69-85, 1997.