# A Memory Markov Chain Model For VBR Traffic With Strong Positive Correlations

O. Rose

Institute of Computer Science, University of Würzburg
Am Hubland, 97074 Würzburg, Germany
Tel.: +49-931-888-5518, Fax.: +49-931-888-4601
Email: rose@informatik.uni-wuerzburg.de
WWW: http://www-info3.informatik.uni-wuerzburg.de/~rose

We propose a new modeling approach for variable bit rate (VBR) traffic in packet networks based on a Markov chain with memory. The model is simple, comprehensive, and facilitates the modeling of strong positive correlations over a considerable range of lags. The model can easily be adapted to measured traffic sequences. In this paper, this is shown for MPEG video traffic and the output of an IS-95 vocoder. Due to its simplicity and its close relationship to Special Semi-Markov Processes (SSMPs) the model is valuable to both simulation and analysis of currently considered types of traffic. We present a discrete-time analysis of buffer with Memory Markov Chain input based on a SSMP/D/1-S analysis.

## 1 Introduction

Currently, the increasing amount of traffic types with strong positive correlations or even long-range dependent behavior in combination with larger buffers in telecommunication equipment demand for traffic models and analytical tools capable to cope with this scenario. For instance, if real-time VBR video traffic is considered as being uncorrelated, performance analysis will underestimate losses and delays at network buffers by orders of magnitude [16, 15].

Thus, there is a demand for stochastic processes facilitating the modeling of such correlation characteristics. In particular, the modeling of VBR video traffic attracted a lot of interest, e.g. see overviews [7, 17]. Among those modeling approaches, there are Markov chains [13], DARs [10], autoregressive processes [1], TES models [11], selfsimilar processes [8], and combinations of these [5]. The simple

modeling approaches, such as histograms or first-order Markov chains, are not capable to provide a good approximation of the autocorrelation function of the empirical data sets. The advanced modeling techniques, such as selfsimilar processes, lead to difficulties in parameter estimation, simulation, and analysis.

We therefore propose a new approach for modeling VBR traffic: the Memory Markov Chain (MMC). In short, the Memory Markov Chain is a modified first-order Markov Chain with enhanced correlation properties. It can be applied for the analysis and simulation of queuing systems with buffers of considerable size. The main idea is to redefine the set of states of a first-order Markov chain. Instead of representing only the sample size, as for a conventional Markov chain, in our model a state represents both the sample size and the average of the sizes of a number of preceding samples. This approach is different from a higher-order Markov chain, since the MMC does not take into account the preceding states but their averaged effect. Thus, we avoid a state space explosion which is typical for higher-order Markov chains. Nevertheless, we are able to improve the correlation behavior considerably.

The parameter estimation of the MMC requires only three parameters: the number of states $M_s$ representing the sample size, the number of samples $W$ which are considered for the averaging process, and the number of states $M_a$ representing the average sizes. Having defined the MMC states, the generation of the transition probability matrix is as straightforward as for a first-order Markov chain. The whole modeling procedure can be automated to find an MMC$(M_s, M_a, W)$ which models the marginal distribution and the autocorrelation function of given data sets over a wide range of lags with high accuracy.

The paper is organized as follows. In Section 2, we define the MMC model, provide its parameter estimation and present a method to find optimal values of $M_s$, $M_a$, and $W$ with little effort. In Section 3, two modeling examples are given: packet video traffic and IS-95 vocoder output traffic. Section 4 is devoted to the analysis of the MMC/D/1-S system. This section includes a number of numerical examples for both the video and voice traffic which show a high degree of modeling accuracy that can be achieved by MMC models.

## 2 The Memory Markov Chain traffic model

### 2.1 Model definition and parameter estimation

Let $\{x_i : i = 1, \ldots, N\}$ denote the considered traffic measurement time series and let $W$ denote the number of samples considered for the averaging process. The given trace $\{x_i\}$ is now extended to a series of pairs $\{(x_i, \bar{x}_i)\}$ with

$$\bar{x}_i \quad = \quad \frac{1}{W} \sum_{k=i-W}^{i-1} x_k \quad \text{for} \quad i = W+1, \ldots, N. \tag{1}$$

The series of pairs consists of $W$ samples less than the original time series.

The averaging process is motivated as follows. For a time series with fast decaying autocorrelation function, the values $\bar{x}_i$ will be rather good estimators of the sample mean and will be almost the same for all values of $i$. Hence, this additional category contains no information and is of very limited value with respect to a better characterization of the sample correlations. However, in the presence of a slowly decaying autocorrelation function, the values $\bar{x}_i$ differ significantly and represent the cumulated and averaged history of $x_i$.

Before we start to determine the transition matrix for the MMC we have to define an appropriate state space. Each pair of the series $\{(x_i, \bar{x}_i)\}$ is related to a state $(m_i^s, m_i^a)$, where $m_i^s \in \{1, \ldots, M_s\}$ denotes the sample size class and $m_i^a \in \{1, \ldots, M_a\}$ the average size class of sample $i$. The classes are obtained by discretizing both the $x_i$ and $\bar{x}_i$ as follows.

$$m_i^s \quad = \quad \begin{cases} 1 & \text{if} \quad x_i \quad = \quad \min_i x_i \\ \left\lceil \dfrac{x_i - \min_i x_i}{\max_i x_i - \min_i x_i} \cdot M_s \right\rceil & \text{otherwise} \end{cases} \tag{2}$$

The $\bar{x}_i$ are processed analogously to obtain $m_i^a$.

The $M_s \cdot M_a$ states are ordered such that the states for one average size class are grouped together with ascending average size class number. The entries of the transition matrix $\mathbf{P}$ can now be estimated as empirical transition probabilities [4]. The size vector $\mathbf{s}$ related to the states of the Markov chain is determined in two steps. First, we compute the size vector of length $M_s$ based on the $x_i$ values.

Let $\delta(m_i^s, j)$ denote the indicator function of $m_i^s = j$ with $j \in \{1, \ldots, M_s\}$. The sizes $s_j$ related to sample size class $j$ are now given by

$$s_j \quad = \quad \frac{\sum_{i=W+1}^{N} \delta(m_i^s, j) \cdot x_i}{\sum_{i=W+1}^{N} \delta(m_i^s, j)}. \tag{3}$$

Then, vector $\mathbf{s}$ is formed by concatenating $M_a$ copies of that vector. For instance, let $M_s = 3$ and $M_a = 2$. Then, the state space consists of $\{(1,1), (2,1), (3,1), (1,2), (2,2), (3,2)\}$ and $\mathbf{s} = [s_1, s_2, s_3, s_1, s_2, s_3]$.

From the above follows that the MMC can be classified by the three parameters $M_s$, $M_a$, and $W$. In the sequel, we will use the notation MMC$(M_s, M_a, W)$ for such a model. A traditional Markov Chain, i.e. a MMC$(M_s, 1, -)$ model, will be denoted by MC$(M_s)$.

The MMC model is fully characterized by the transition matrix $\mathbf{P}$ and the size vector $\mathbf{s}$. The mean and variance of the process generated by this model are given by

$$\mu \quad = \quad \boldsymbol{\pi} \cdot \mathbf{s}^T, \qquad \sigma^2 \quad = \quad \sum \boldsymbol{\pi} \cdot diag(\mathbf{s})^2 - \mu^2 \tag{4}$$

where the vector $\boldsymbol{\pi}$ is the steady-state distribution of the MMC given by the solution of

$$\boldsymbol{\pi} \quad = \quad \boldsymbol{\pi}\mathbf{P}, \qquad \sum \boldsymbol{\pi} = 1. \tag{5}$$

and $diag(\mathbf{s})$ denotes a square diagonal matrix with diagonal $\mathbf{s}$. The correlation coefficient for lag $k$ is obtained by

$$\rho_k \quad = \quad \frac{1}{\sigma^2}\left[\sum \boldsymbol{\pi} \cdot diag(\mathbf{s}) \cdot \mathbf{P}^k \cdot diag(\mathbf{s}) - \mu^2\right]. \tag{6}$$

Note, that a sequence generated by means of transition matrix $\mathbf{P}$ and the size vector $\mathbf{s}$ will have a marginal distribution which is identical to a histogram with $M_s$ intervals of equal size.

### 2.2 Estimation of $M_a$ and $W$

We assume that the value $M_s$ is given, for instance, due to a desired level of accuracy for the marginal distribution of the MMC. The values $M_a$ and $W$ have to be determined empirically. To automate this procedure, we suggest to use the following method. First, a range of lags $0 \ldots l_{max}$ is defined for which the model autocorrelation function (ACF) matches the sample ACF as well as possible. The distance of model ACF $\rho_l(M_s, M_a, W)$ and sample ACF $\hat{\rho}_l$ may be measured by

$$\Delta(M_s, M_a, W) \quad = \quad \frac{1}{l_{max}} \sum_{l=1}^{l_{max}} (\rho_l(M_s, M_a, W) - \hat{\rho}_l)^2 \tag{7}$$

Then, the minimum of $\Delta(M_s, M_a, 10 \cdot k), k = 1, 2, \ldots$ for a given value of $M_a > 2$ is determined. The values $W = 10 \cdot k$ were chosen to reduce computation time. Our experiments showed that $W = 10 \cdot k$ should be at most equal to $l_{max}$. This procedure is now iterated for increasing values of $M_a$ until both ACFs match sufficiently well. Tables 1 and 3 show the results of this method for the video and the voice data sets presented in the next two sections.

The optimal values of $M_a$ and $W$ provide a starting point for the physical interpretation of the model. For instance, if video traces are modeled the value $M_a$ can be seen as the number of scene classes and the corresponding optimal value of $W$ as the average length of a scene. If voice traces are considered a value of $M_a$ which is larger than 2 provides the information that a simple talkspurt/silence model may not be adequate to model the voice source.

## 3 Modeling examples

### 3.1 Packet video traffic

To show the usefulness and the validity of our approach, we applied the MMC as a model for variable bit rate packet video. As video data set, we chose the well-known Bellcore *Star Wars* sequence [8]. Details on packet video and statistical properties of video data sets can be found in the related literature,

e.g. [14]. The original Bellcore sequence provides the frame sizes in bits which we transformed into a sequence of numbers of ATM cells assuming a payload of 48 bytes.

For the remainder of this section, we assume a number of states $M_s = 20$ for the MMC which is a good compromise with respect to accuracy of the MMC's marginal distribution and the total number of MMC states. Figure 1 depicts the complementary probability distribution function (CPDF) of the frame sizes of the original trace and an MMC$(20, M_a, W)$ model.

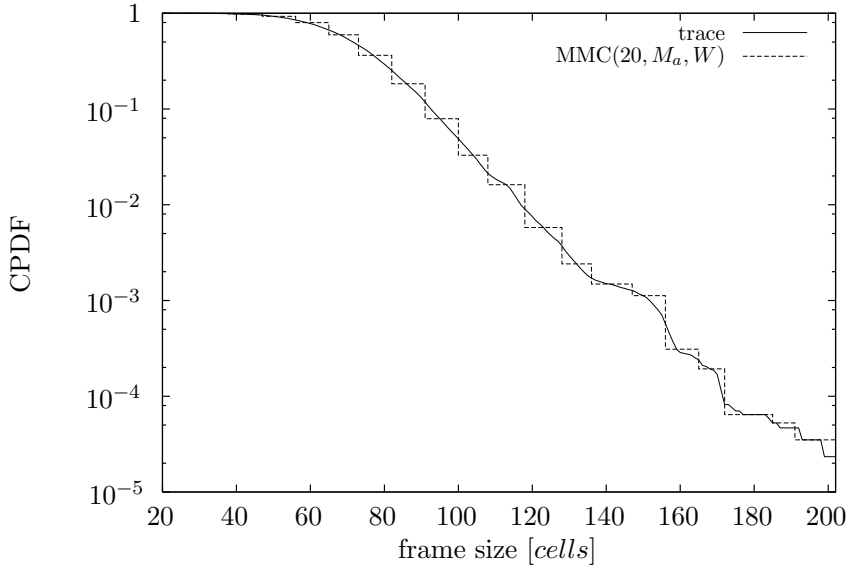


Figure 1: Cumulative probability distribution function of the video trace and model

Table 1 shows the optimal window sizes $W_{opt}(M_a)$ for a number of $M_a$ values and the values of $\Delta(20, M_a, W_{opt}(M_a))$. The objective is to find a MMC ACF which matches the sample ACF as well as possible for the first 200 lags. For details see Section 2.2.

Table 1: Optimal window sizes of the video MMC

| $M_a$ | **1** | **2** | **5** | **10** | **15** | **20** |
|---|---|---|---|---|---|---|
| $W_{opt}$ | — | **300** | **100** | **100** | **60** | **60** |
| $\Delta(20, M_a, W_{opt})$ $[10^{-3}]$ | **149.3** | **92.0** | **34.8** | **11.3** | **5.3** | **2.3** |

Figure 2 shows the ACFs of the video trace and of several MMCs. In addition, we provide the ACF of a *Fractional Gaussian Noise (FGN)* process with a Hurst parameter of $H = 0.8$. This Hurst parameter is reported for the example video sequence by the authors of [8]. Methods to estimate the Hurst parameter and details on fractal processes such as FGN can be found in e.g. [3].

The figure reveals that a large number of $M_a$ states, say at least 10, is required to obtain a good

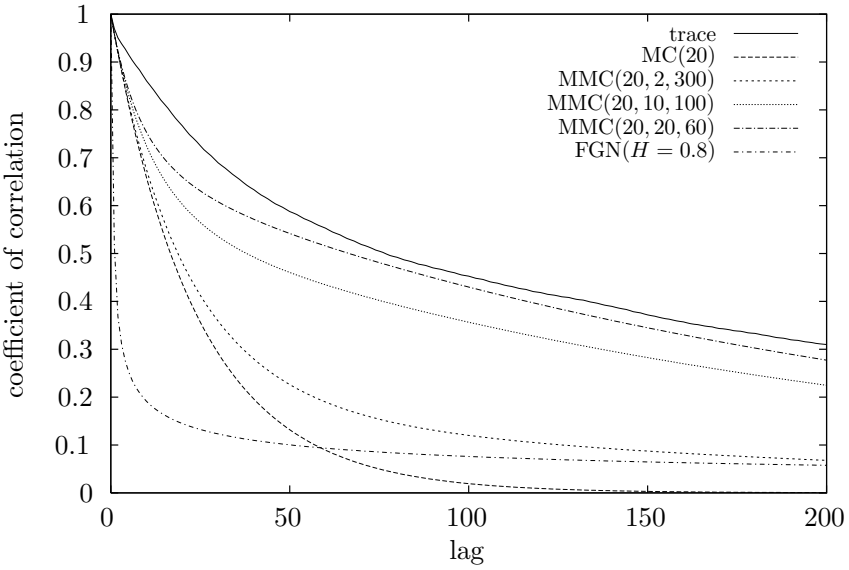


Figure 2: Autocorrelation functions of video MMCs

MMC approximation of the video ACF for the first 200 lags. A traditional Markov Chain, i.e. the MC(20) model, is far from being capable to model the sample ACF. The same is true, however, for the FGN model of the video data set. For the first few hundred lags a simple fractal process is not capable to provide a realistic picture of the sample correlations. If such a model is applied to assess the performance of a buffer of moderate size with video traffic input the result will be performance estimates such as cell losses or delays which are too optimistic. This indicates that for moderate buffer sizes well-designed MMC models clearly outperform simple fractal models with respect to accuracy of performance estimates.

### 3.2 IS-95 vocoder traffic

As a second example, we use MMCs to model the output of a voice encoder which is used for the compression of digitized audio for internet and mobile telephony. For compressing our audio trace we chose a variable data rate vocoder software as specified in Appendix A of the IS-95 standard on mobile station – base station compatibility for dual mode wideband spread spectrum cellular systems [19].

Depending on voice activity, the vocoder provides four different data rates, namely 9600 bps (full rate), 4800 bps (1/2 rate), 2400 bps (1/4 rate), or 1200 bps (1/8 rate). This is realized by emitting data packets of length 171, 80, 40, or 16 bits every 20 ms.

As vocoder input we used an audio track from a CD recorded by the Bavarian Archive for Speech Signals which contains a number of standard telephone dialogues [2]. The considered track (Q008N A) contains the calling person's part of a hotel reservation telephone dialogue. The dialogue is held in

English by male persons and lasts about 7 min. We obtained a vocoder output data stream of 21160 packets from which we extracted a sequence of vocoder states ranging from 1/8 rate to full rate. This trace is now used for modeling.

Figure 3 shows the first thousand samples of this trace, i.e. 20 s of the dialogue. As long as there is high voice activity the vocoder emits packets at full rate. As soon as a sentence ends the vocoder drops to 1/8 rate for a short time. As long as the customer listens to the person at the hotel reception the vocoder remains on the lowest data rate for a longer period (for instance, samples 220 to 400).

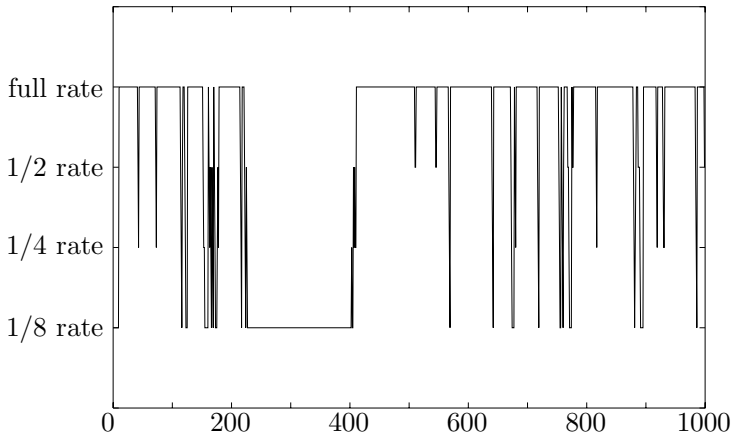


Figure 3: Output rates of the IS-95 vocoder

Table 2 reports the relative frequency of the vocoder states.

Table 2: Relative frequencies of the vocoder states

| State | Number | Rel. Frequency |
|---|---|---|
| **Full rate** | **11638** | **0.550** |
| **1/2 rate** | **1025** | **0.048** |
| **1/4 rate** | **736** | **0.035** |
| **1/8 rate** | **7761** | **0.367** |
| **Total** | **21160** | **1.000** |

For the MMC model, the number of states $M_s$ is set to 4 representing the 4 vocoder data rates. For the number of average size states $M_a$ and the optimal window size $W_{opt}(M_a)$, an optimization is used as outlined in Section 2.2. The objective is to find a MMC ACF which matches the sample ACF as well as possible for the first 200 lags.

Assuming a maximum ACF lag of $l_{max} = 200$, Table 3 shows the optimal window sizes for a number of $M_a$ values, and the average squared error of the model ACF with respect to the sample ACF.

Table 3: Optimal window sizes $W$ of the MMC

| $M_a$ | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| $W_{opt}$ | — | **70** | **50** | **50** | **40** | **30** | **30** | **30** |
| $\Delta(4, M_a, W_{opt})$ $[10^{-4}]$ | **1469.8** | **95.6** | **12.1** | **4.2** | **2.1** | **1.2** | **1.6** | **2.6** |

The results show that the MMC with $M_a > 1$ is superior to an MC model with respect to ACF modeling capabilities. The optimal modeling performance is obtained for MMC$(4, 6, 30)$, i.e. a model which needs 24 states. In contrast to the video case, the error does not monotonically decrease with the parameter $M_a$. For parameters larger than $M_a = 6$, the error becomes larger again.

Figure 4 shows the ACFs of the sample trace and of a number of MMCs.

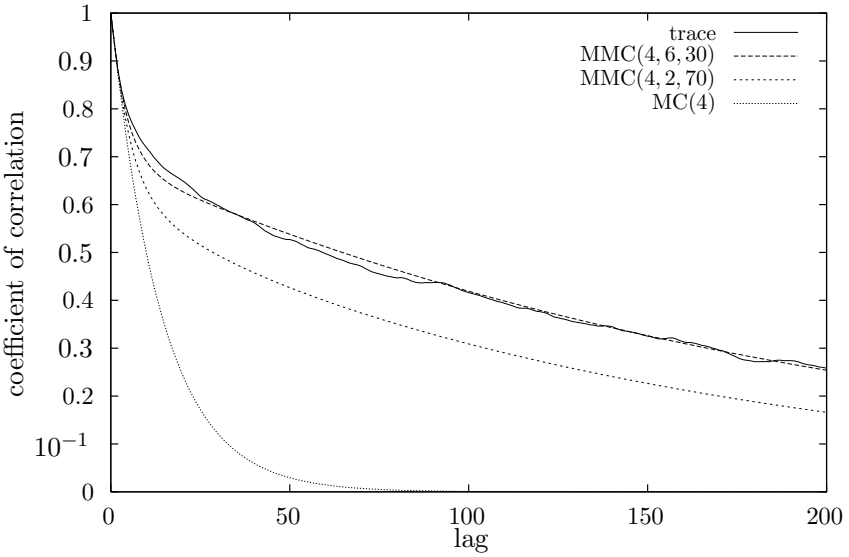


Figure 4: Autocorrelation functions of the voice MMCs

## 4 Analysis of the MMC/D/1-S system

For the analysis of the MMC/D/1-S system, we modify the discrete-time analysis of the SSMP/G/1 system presented by Ding and Decker [6] which was extended to SMP/G/1-S systems by Herrmann [9]. The MMC is the most simple form of a Special Semi-Markov Process (SSMP) since the sojourn time distributions are reduced to a single value in this case. This considerably simplifies the analysis and speeds up the computation of performance measures.

In contrast to the referenced publications, we use the MMC to model the amount of data arriving in a fixed period of time, and not interarrival times of data packets. In the sequel, this period of time

is named frame duration. This change in modeling perspective is due to the fact that the data traffic we model in our example is emitted by encoders which generate data frames at a constant rate. For instance, the vocoder emits data frames of varying size every 20 ms.

If varying interarrival times of data packets should be modeled by a MMC, the referenced models need not to be modified.

## 4.1 System model

For our analysis, we consider a one-stage queuing system with a single MMC traffic source. The buffer is of finite length and the service time is constant. Figure 5 depicts the system and provides the most important system parameters. For system analysis, we make the following assumptions: The time is

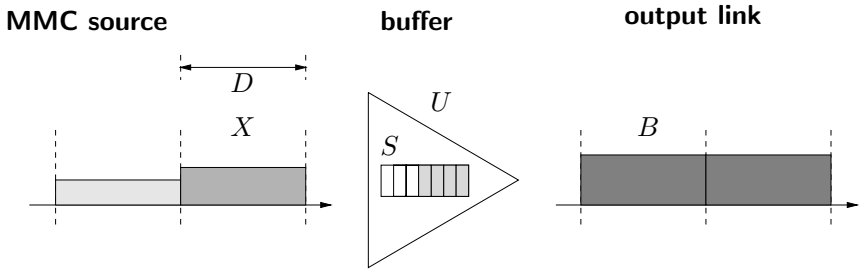


Figure 5: Buffer with a single MMC traffic source

discretized into frame durations $D$, i.e., the reciprocal of the frame rate, and synchronized to frame starts.

Data packets of a single frame are regarded as a fluid according to the *fluid simulation approach* [7]. Instead of individual packet arrivals we consider frame data as a fluid, which flows into the buffer at a constant rate. There are two important benefits from the fluid approach. It is conceptually simple and it leads to a reduced algorithmic complexity while the accuracy of the results is comparable to data unit-oriented approaches. Loss probabilities can be calculated in terms of overflow volumes.

In case of video traffic, the fluid approach is also useful from practical view point. In [18], the authors report that smoothing the data units of a video source to a constant data rate during one frame duration outperforms all other data emission schemes in terms of multiplexer performance.

## 4.2 Analysis

As an introduction to the analysis of the correlated input MMC/D/1-S model, we first present the discrete-time analysis of the above system with uncorrelated input traffic. The following notation is used to characterize the system:

$X_n$ : random variable for the amount of data in the $n$th frame,

$B$ : random variable for the number of data units transmitted during one frame duration $D$; in our case $B$ is deterministic and we denote the constant number of data units transmitted by $b$,

$U_n$ : random variable for the number of data units waiting in the buffer upon the arrival of the $n$th frame, the socalled unfinished work,

$S$ : buffer size.

The random variables $X_n$, $B$, and $U_n$ follow discrete distributions (probability mass functions) $x_n(k)$, $b(k)$ and $u_n(k)$, where, for instance, $x_n(i) = \Pr\{X_n = i\}$.

Due to our fluid-flow assumption, we are able to simplify the computation of the distribution of the buffer content $u_n(k)$ considerably as compared to standard unfinished work approaches such as, e.g., in [20]. The system evolution is determined by the following equation:

$$U_{n+1} \quad = \quad \min\left\{\max\{U_n + X_n - b, 0\}, S\right\}. \tag{8}$$

At first glance, our approach looks like a typical batch arrival process with batch size of $X_n$. In contrast to the original method, however, we already subtract the maximum amount of data which can be transmitted during a frame duration $b$ upon arrival of frame $n$. This has to be done due to the fluid-flow assumption and the discretization of the system time into frame durations. After the arrival of one frame as a batch of data units, we have to subsume in one equation the behavior of the modeled system until the next frame arrival, i.e., equally spaced data units entering a buffer which is served at a constant rate. Integrating both arrival and service process over $D$, and taking into account the content of the buffer before the frame arrival as well as the limited buffer size leads to Equation (8). If the buffer content plus the new data minus the amount of data which can be transmitted exceeds the buffer size then data units are lost. If more data can be transmitted than the buffer content plus arriving data then the buffer runs empty. In all other cases the buffer contains data units and there are no losses during the frame duration.

Equation (8) leads to the following recursion for the distribution of the buffer content:

$$u_{n+1}(k) \quad = \quad \pi^S \circ \pi_0 \left[u_n(k) \circledast x(k) \circledast \delta(k+b)\right]. \tag{9}$$

Here, $\circledast$ denotes the discrete convolution, $\pi_0$ and $\pi^S$ are sweep operators, and $\delta(k+b)$ is the shifted Kronecker delta. The operators are defined as follows.

- *Discrete convolution.*

$$c(k) \quad = \quad a(k) \circledast b(k) \quad = \quad \sum_{j=-\infty}^{\infty} a(k-j) \cdot b(j). \tag{10}$$

- *Low sweep operator.*

$$\pi_0[a(i)] \quad = \quad \begin{cases} 0 & : \quad i < 0, \\ \sum\limits_{j=-\infty}^{0} a(j) & : \quad i = 0, \\ a(i) & : \quad i > 0. \end{cases} \tag{11}$$

- *High sweep operator.*

$$\pi^S[a(i)] \quad = \quad \begin{cases} a(i) & : \quad i < S, \\ \sum\limits_{j=S}^{\infty} a(j) & : \quad i = S, \\ 0 & : \quad i > S. \end{cases} \tag{12}$$

- *Kronecker delta.*

$$\delta(i) \quad = \quad \begin{cases} 1 & : \quad i = 0, \\ 0 & : \quad \text{otherwise.} \end{cases} \tag{13}$$

Assuming that the buffer content distribution of the multiplexer converges to a steady state, $u(k) = \lim_{n\to\infty} u_n(k)$ is the solution of the fix-point equation

$$u(k) \quad = \quad \pi^S \circ \pi_0 \left[ u(k) \circledast c(k) \right], \tag{14}$$

where

$$c(k) \quad = \quad x(k) \circledast \delta(k+b). \tag{15}$$

Equation (14) is the discrete-time analogue of the Lindley integral equation of $GI/G/1-S$ queuing systems (see [12]).

In the following, the basic algorithm is extended to an MMC arrival process. To this end, the notation is updated as follows:

| | | |
|---|---|---|
| $X_{n,i}$ | : | random variable for the amount of data in the $n$th frame while the MMC is in state $i$, |
| $U_{n,i}$ | : | random variable for the amount of data waiting in the buffer upon the arrival of the $n$th frame while the MMC is in state $i$, |
| $U^v_{n,i}$ | : | random variable for the virtual amount of data waiting in the buffer, |
| $U^-_{n,i}$ | : | random variable for the actual amount of data waiting in the buffer immediately before the state transition of the MMC, |

It is assumed that the state transition takes place immediately before updating the buffer content. In the following, $i = 1, \ldots, M_s \cdot M_a$ denotes a state of the MMC ,$k \in \mathbb{Z}$ the element index of a distribution, and $n = 0, 1, 2, \ldots$ the iteration counter.

Right after the arrival of MMC data the virtual buffer content is distributed as

$$\begin{aligned} u^v_{n,i}(k) &= u_{n,i}(k) \circledast x(k) \circledast \delta(k+b) && (16)\\ &= u_{n,i}(k) \circledast \delta(k-s_i) \circledast \delta(k+b) && (17)\\ &= u_{n,i}(k) \circledast \delta(k+b-s_i) && (18)\\ &= \sigma^{s_i-b}\left[u_{n,i}(k)\right]. && (19) \end{aligned}$$

where $\sigma^l$ is a shift operator which moves the elements of a distribution $l$ positions to the right. This leads to an actual state dependent buffer content distribution of

$$u^-_{n,i}(k) = \pi^S \circ \pi_0 \left[u^v_{n,i}(k)\right]. \quad (20)$$

The buffer content distribution right after the state transition and just before the arrival of new data is now given by

$$u_{n+1,j}(k) = \sum_{i=1}^{M_s \cdot M_a} p_{ij} u^-_{n,i}(k). \quad (21)$$

Iterating on this set of equations leads to a set of state dependent buffer content distributions $u_i(k) = \lim_{n\to\infty} u_{n,i}(k)$.

The unconditioned distribution of the buffer content is provided by

$$u(k) = \sum_{i=1}^{M_s \cdot M_a} u_i(k). \quad (22)$$

Since loss of data occurs if the virtual buffer content is larger than buffer size $S$ the loss probability of the system reads

$$P_{loss} = \frac{\sum_{i=1}^{M_s \cdot M_a} \sum_{k=S+1}^{S+s_i+1} u^v_i(k)(k-S)}{\mu} \quad (23)$$

where $\mu$ denotes the average amount of data per frame.

### 4.3 Numerical results

The above MMC/D/1-S analysis is now used to estimate data unit losses of a single video or voice source at a buffer with a constant output rate. We consider a wide range of buffer sizes and system loads to show the estimation capabilities of a variety of MMCs. To adapt the system to a given load $\rho$, the amount of data flowing out of the buffer during one frame duration is set to $\lceil \mu/\rho \rceil$.

The loss results for the measured data sequences are obtained by means of trace-driven simulations.

In this section, we focus on the video MMC models. Figure 6 shows the cell loss results for a buffer of 1000 cells. For loads larger than 0.55, the MMC$(20, 20, 60)$ provides almost the same cell losses as the original trace, whereas a standard MC and a histogram model considerably underestimate the

cell losses. All three models have the same marginal distribution as depicted in Figure 1. Hence, the differences in loss estimation accuracy only result from the different correlation modeling capabilities of the models.

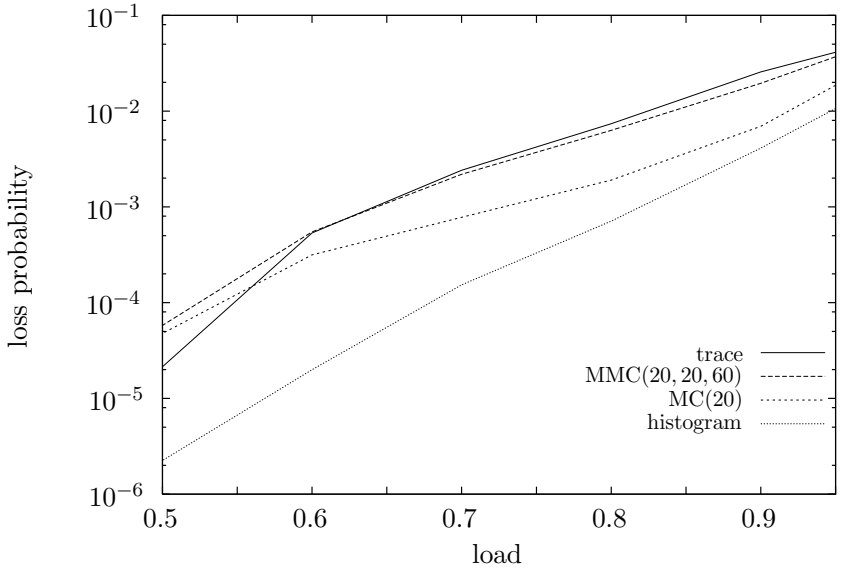


Figure 6: Video data losses for a buffer size of 1000 cells

Figure 7 depicts the cell losses for a load of 0.95. As in the previous graph, the MMC provides a high accuracy in estimating the cell losses even in the case of large buffers. In comparison, the standard Markov chain underestimates the cell losses of the trace-driven simulation by up to several orders of magnitude.

## 5 Conclusion

In this paper, we presented the Memory Markov Chain (MMC) as a new model for strongly positively correlated data traffic. The model is simple, comprehensive, and lends itself to standard analysis techniques. The model is applied to both video and voice traffic. For the given sample traces, well-designed MMC models provide in both cases very good approximations of the marginal distribution and the autocorrelation function over the first few hundreds of lags.

In the second part of the paper, we outlined the exact analysis of a discrete MMC/D/1-S system with a focus on the estimation of the loss probability at the system buffer. A large variety of numerical results indicate that the MMC is superior to standard Markov Chain or histogram models which provide performance estimates that are too optimistic for strongly correlated traffic.

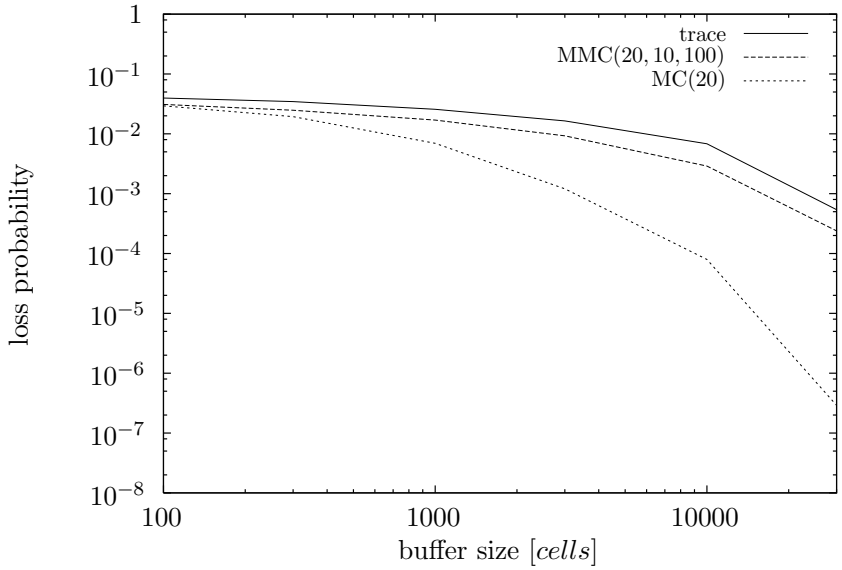


Figure 7: Video data losses for a load of 0.9

## References


[1] A. Adas. Supporting real time VBR video using dynamic reservation based on linear prediction. In *Proceedings of the Infocom '96*, pages 1476–1483, 1996.

[2] Bavarian Archive for Speach Signals (BAS). Verbmobil 6.1, 1996. Further information via http://www.phonetik.uni-muenchen.de.

[3] J. Beran. *Statistics for long-memory processes*. Chapman & Hall, New York, 1994.

[4] P. Billingsley. *Statistical Inference for Markov Processes*. University of Chicago Press, Chicago, 1961.

[5] K. Chandra and A. R. Reibman. Modeling two-layer SNR-scalable MPEG-2 video traffic. In *Proceedings of the 7th International Workshop on Packet Video*, Oct. 1996.

[6] W. Ding and P. Decker. Waiting time distribution of a discrete SSMP/G/1 queue and its implications in ATM systems. In *Proceedings of the 7th ITC Specialist Seminar*, page 9.4, Oct. 1990.

[7] V. S. Frost and B. Melamed. Traffic modeling for telecommunication networks. *IEEE Communications Magazine*, 32(3):70–81, Mar. 1994.

[8] M. W. Garrett and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proceedings of the ACM SIGCOMM '94 Conference*, pages 269–280, 1994.

[9] C. Herrmann. Analysis of the discrete-time SMP/D/1/s finite buffer queue with applications in ATM. In *Proceedings of the IEEE Infocom '93*, pages 160–167, Mar. 1993.

[10] D. P. Heyman, A. Tabatabai, and T. V. Lakshman. Statistical analysis and simulation study of video teleconference traffic in ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2(1):49–59, Mar. 1992.

[11] M. R. Ismail, I. E. Lambadaris, M. Devetsikiotis, and A. R. Kaye. Modelling prioritized MPEG video using TES and a frame spreading strategy for transmission in ATM networks. In *Proceedings of the Infocom '95*, pages 762–770, 1995.

[12] L. Kleinrock. *Queueing Systems – Volume 1: Theory.* Wiley, New York, 1975.

[13] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Transactions on Communications*, 36(7):834–844, July 1988.

[14] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In *Proceedings of the 20th Annual Conference on Local Computer Networks*, pages 397–406, Oct. 1995.

[15] O. Rose. Discrete-time analysis of a finite buffer with VBR MPEG video traffic input. In *Proceedings of the ITC-15*, pages 413–422, 1997.

[16] O. Rose. Simple and efficient models for variable bit rate MPEG video traffic. *Performance Evaluation*, 30:69–85, 1997.

[17] O. Rose and M. R. Frater. A comparison of models for VBR video traffic sources in B-ISDN. In *IFIP Transactions C-24: Broadband Communications, II*, pages 275 – 287. North-Holland, Amsterdam, 1994.

[18] P. Skelly, M. Schwarz, and S. Dixit. A histogram-based model for video traffic behavior in an ATM multiplexer. *IEEE/ACM Transactions on Networking*, 1(4):446–459, Aug. 1993.

[19] Telecommunications Industry Association. *TIA/EIA/IS-95. Mobile Station – Base Station Compatibility Standard for Dual Mode Wideband Spread Spectrum Cellular Systems*, 1993.

[20] P. Tran-Gia and H. Ahmadi. Analysis of a discrete-time $G^{[X]}/D/1-S$ queueing system with applications in packet-switching systems. In *Proceedings of the Infocom '88*, pages 861–870, 1988.