

# Performance Evaluation of Video Traffic Models

Wassim Abbessi

National School for Computer Science,  
University of Manouba,  
Tunisia

Hédi Nabli

King Faisal University, College of Science,  
Department of Mathematics & Statistics,  
P.O. 380 Ahsaa 31982, Kingdom of Saudi Arabia

**Abstract**—This paper focuses on the performance evaluation of three models for video sources. We compare the statistical characteristics of network traffic of a set of video sources with those of artificial traffic generated using the models. The results indicate that the fluid Markov model offers the best performance and a good tradeoff between accuracy and simplicity.

**Keywords**— Video traffic; GoP-based model; performance evaluation

## I. INTRODUCTION

Developing mathematical models for network traffic is an essential tool to understand and predict the network behavior facing various types of traffic. As the end-user needs grow and change quantitatively and qualitatively through time, the need for new models is still present. In the last decade, with the emergence of wireless networks and mobile devices, operators and service providers had to give more and more resources to video traffic. Trend reports [1] show that the various types of video traffic (IPTV, VoD, mobile video, UHD, etc.) are highly impacting the traffic characteristics in the networks.

Compared to classical network traffic (web, mail and chat), the video traffic offers some particular constraints and challenges: In one hand, it requires more bandwidth than most of the other network traffic sources. On the other hand, it has stringent delay and loss requirements. That is why the transport of video traffic over the networks requires specific traffic engineering policies and adequate models.

In the literature, we find a large number of contributions related to modeling video traffic (see [2]–[4], [6] for a survey). Depending on the mathematical tools used, the models can be classified into categories: We find auto-regressive (AR), Markov, self-similar, wavelet and scene-based models. Most of these models are specific to one particular type of codecs (MPEG-1 for example). They cannot be applied to other types of video encoding without important changes. Though, on the video market we find a large set of video encoding techniques that are evolving constantly and that can support the different customers' needs. For this reason, one of the criteria to choose a suitable model for video source traffic is to support many encoding technologies and to be able to support new ones without significant changes.

In [5], we defined a fluid Markov model for video traffic. This model that we will denote by GBFM (GoP-based fluid model) is based on classifying video GoPs (Group of Pictures) in scenes and modeling each scene with a phase-type Markov process. One of the main advantages of this model is that it can be applied to various encoded video types. In this paper, we focus on the performance evaluation of the model and compare it with other reference models from various

categories. We chose to compare it with a DAR(1) model because DAR models are very common in literature and known for their simplicity. Since our model is based on Markov processes, we compared it also with one of the most cited Markov models. In order to have trustworthy results, we respected the evaluation methodology presented in [6]. We show that while using classical and well-known tools such as Markov models, the model predicts accurately the statistical distributions in network buffer and generates artificial traffic that has the same statistical properties than the original traffic trace.

This paper is organized as follows: In section II, we describe the GBFM model and the other models to be compared with. In section III, we detail our evaluation results. The conclusions are given in section IV.

## II. VIDEO TRAFFIC MODELS

In this section, we give a brief description of the models that will be compared in the next section.

### A. The GBFM model

First, we describe the GoP-based Fluid Model as defined in [5]: A continuous-time Markov process  $(X_t)_{t \geq 0}$  models the evolution of the throughput of a video source through time. The Markov process has  $n$  states. The process is fully known by specifying its infinitesimal generator  $\mathbf{A} = (a_{ij})_{i,j \in S}$  and its initial distribution  $\boldsymbol{\alpha} = (P\{X_0 = i\}, i \in S)$ .

The video source is connected to a network buffer with finite or infinite capacity. The buffer's output rate evolves also dynamically with the Markov process  $(X_t)_{t \geq 0}$ . When the video source is in the state  $i \in S = \{1, \dots, n\}$ , it generates video data at a constant fluid throughput  $r_i$  and the buffer's throughput is denoted  $c_i$ . The value  $d_i = r_i - c_i$  corresponds to the effective input rate associated to the state  $i \in S$ . The buffer content at instant  $t$  is denoted  $Q_t$ . The diagonal matrix  $\text{diag}(d_i, i \in S)$  is denoted by  $\mathbf{D}$ .

The model parameters are then: The infinitesimal generator  $\mathbf{A} = (a_{ij})_{i,j \in S}$ , the drift matrix  $\mathbf{D}$  and the initial distribution  $\boldsymbol{\alpha} = (P\{X_0 = i\}, i \in S)$ .

Now, we detail how these parameters are inferred from the trace of the video source. First, we use the original video trace to extract the sizes of the video frames. We compute GoP sizes and then GoPs are classified into  $m$  classes using the global k-means classification algorithm [7]. We associate to each class of GoPs a *macrostate* of the process  $(X_t)_{t \geq 0}$ . These macrostates are denoted  $M_i, i \in \{1, \dots, m\}$ . A

macrostate is an intermediate virtual state that will correspond later to a subset of states of the Markov process.

The GoPs are then regrouped to create *scenes*. Each scene is a set of adjacent GoPs belonging to the same class (associated to the same macrostate). The model assumes that for all scenes of the same class the video source throughput is constant and is equal to the average size of all GoPs belonging to that class. For each macrostate  $M_i$ , the video throughput is constant and equal to  $r_i$ . Knowing the output rate  $c_i$  (which is generally constant), the drift matrix  $\mathbf{D}$  is therefore entirely determined. The transition rates between the macrostates can be obtained using the transition rate matrix denoted  $\mathbf{P} = (p_{ij})_{i,j \in \{1, \dots, m\}}$  associated to the macrostate process. The coefficient  $p_{ij}$  represents the transition probability from the macrostate  $i$  to the macrostate  $j$ . It is calculated proportionally to the transitions observed in the trace after regrouping the GoPs into scenes.

For each class of scenes, the set of scene lengths is fit with a phase-type Markov process. The parameters of this process are obtained via a fitting algorithm. The phase-type process has  $p$  states and an absorbing state denoted state 0. The empirical distribution of scene length is fitted with the distribution of the time to reach the absorbing state.

The global Markov process combines phase-type processes with macrostates. It is obtained by replacing the macrostates in the Markov process with their respective equivalent phase-type processes built using the fitting technique. In other terms, the phase-type processes that capture the statistical properties of the remaining in one class of scenes are modulated by the macrostates process built using the observation of scenes in the original trace. The initial distribution is simple: we suppose that we start always from the class of the first GoP in the original trace.

Each macrostate  $M_i$  of a class  $i$  has an associated phase-type process denoted  $(Y_t^{(i)})_{t \geq 0}$  having  $p$  states. Each phase-type process is characterized by an initial distribution  $\alpha^{(i)}$  and infinitesimal generator  $\Lambda^{(i)} = (\lambda_{jk}^{(i)})_{j,k \in \{0, \dots, p\}}$  where the state 0 corresponds to the absorbing state.

The global Markov process  $(X_t)_{t \geq 0}$  has  $n=m.p$  states denoted  $(i,j)$  for the state  $j$ ;  $1 \leq j \leq p$ ; of the phase-type process  $(Y_t^{(i)})$  where  $i \in \{1, \dots, m\}$ . The transition rate from the state  $(i,j)$  to the state  $(k,l)$  is denoted  $\mu_{ijkl}$ . We recall that the probability matrix between the macrostates is  $\mathbf{P} = (p_{ij})_{i,j \in S}$ . We have then  $\forall i, k \in \{1, \dots, m\}; \forall j, l \in \{1, \dots, p\}$

$$\mu_{ijkl} = \begin{cases} \lambda_{jl}^{(i)} & \text{if } k = i \\ \lambda_{j0}^{(i)} \cdot p_{ik} \cdot \alpha_l^{(k)} & \text{if } k \neq i \end{cases} \quad (1)$$

Now that all the parameters of the global Markov process have been specified, the video source is fully characterized by the infinitesimal generator  $\mathbf{A} = (a_{ij})_{i,j \in S}$ , the drift matrix  $\mathbf{D}$  and the initial distribution  $\alpha = (P\{X_0 = i\}, i \in S)$ .

The artificial traffic generated using this model walks through the global Markov process and generates random size GoPs using the Johnson SB distribution. For each state of the Markov process, the parameters of the distribution are obtained from the statistics of the original trace. Then For each GoP, the I,B and P frames are generated with respect to the ratio I/P and P/B observed in the trace. For more mathematical details about the GBFM model, see [5].

### B. The discrete auto-regressive model DAR(1)

A discrete auto-regressive model of order  $p$ , denoted  $DAR(p)$  generates a sequence of values obtained by a weighted linear combination of past values given by the expression:

$$x(n) = \sum_{i=1}^p a_i x(n-i) + e(n)$$

where  $a_1, a_2, \dots, a_p$  are the AR coefficients. The sequence  $e(n)$  consists of i.i.d. random variables having an arbitrary distribution. The residuals  $e(n)$  are chosen to match the mean and the variance of the original data.

When modeling video traffic using a DAR process,  $x(n)$  represents the size of the  $n^{\text{th}}$  frame of the video. The parameters  $a_i, i = 1, \dots, p$  represent the lag  $i$  auto-correlation of the successive frame sizes.

In this paper, we will use a  $DAR(1)$  process as described in [8]. For each type of frames (I,B and P), a  $DAR(1)$  is used to model frame sizes. A  $DAR(1)$  process is equivalent to a Markov chain with a state space  $S$  and a transition matrix:  $P = \rho \mathbf{I} + (1-\rho)\mathbf{Q}$ , where  $\rho$  is the lag-1 auto-correlation coefficient, and  $\mathbf{I}$  is the identity matrix.

The  $\mathbf{Q}$  matrix consists of the Pearson type V probabilities  $\{f_0, f_1, \dots, f_k, F_K\}$ , where  $F_k = \sum_{k > K} f_k$  and  $K$  is the peak rate. Each  $k, k < K$ , corresponds to a possible source rate less than the peak rate  $K$ .

Using the frame statistics obtained from the video trace, the frame sizes are then generated using a residual with the Pearson V distribution with parameters  $(\alpha, \beta)$  given below. The probability distribution function (pdf) of the Pearson V distribution is:

$$f(x) = \frac{x^{-(\alpha+1)} e^{-\beta/x}}{\beta^{-\alpha} \gamma(\alpha)} \quad (2)$$

$$\text{where Mean} = \frac{\beta}{\alpha - 1} \text{ and Variance} = \frac{\beta^2}{(\alpha - 1)^2 (\alpha - 2)}$$

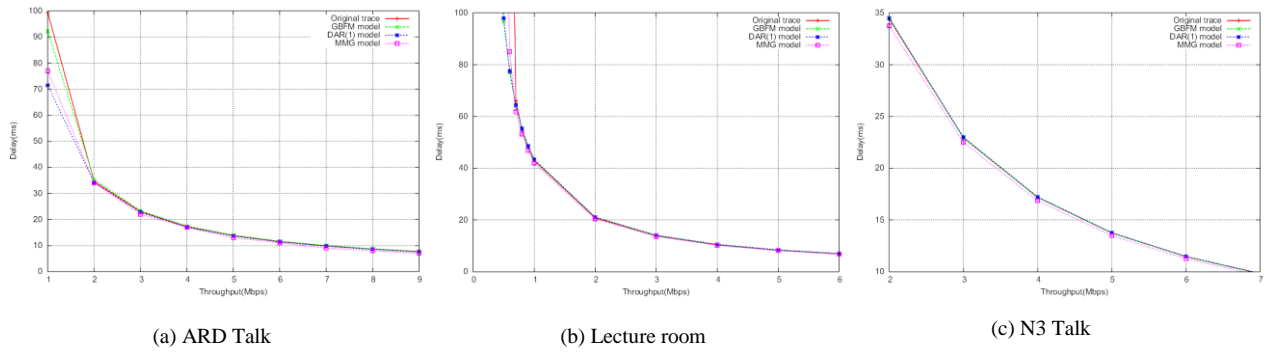


Fig. 1. Delay for original and artificial traffic

The **P** matrix is then generated using the **Q** matrix and the lag-1 auto-correlation coefficient. If the current frame has a size of *i*, then the next frame will have the same size *i* with probability  $\rho + (1 - \rho)f_i$ , and *k* cells;  $k \neq i$ , with probability  $(1 - \rho)f_k$ .

To generate artificial traffic, the model starts from a random frame size and generates frame sizes while using the transition probability matrix **P** until the required number of frames is generated. The I, P and B frames are generated separately using their respective transition probability matrices and then multiplexed according to the required GOP format.

Modeling video using *DAR*(1) process is quite simple since it requires few parameters but the correlation between frames of the same GoP is not taken into consideration since the I, B and P frames generation processes are independent.

*C. The Markov-modulated GAMMA model*

As an example of Markov modulated processes, we will describe the Markov modulated Gamma model [9]. In this model, the video GoPs organized in *clips*. A clip is a set of GoPs of similar size. When a GoP has a significantly different size, a new clip begins. The clips are then sorted in *n* shots depending on their average GoP size. The sizes of shot intervals have a geometric progression. Let us denote by *a* the size of the smallest GoP and *b* is the size of the largest GoP. The shot intervals are  $[a, ar], [ar, ar^2], \dots, [ar^{n-1}, b]$  where  $r = e^{(\ln(b) - \ln(a))/n}$  which can be simply written as  $r = (b/a)^{1/n}$ . The whole video is then partitioned into *n* shots. The authors recommend using  $n = 7$  for optimal performance.

A Markov process is defined to model transition between shots. The transition probability matrix **P** is obtained from the succession of shots:  $P_{ij} = (\text{number of times clip } j \text{ follows a clip } i) / (\text{number of clips } i)$ . After that, a set of  $3 \times n$  parameters of a gamma distribution are computed to model the I, P and B frames in each shot. The durations of clips in one shot follow a gamma distribution with specific parameters.

The main advantage of this model is that it combines information concerning GoPs and frames but it depends on the choice of the threshold values that may affect the clustering algorithm. Furthermore, the number of parameters

that the model requires is important. In addition, the model eliminates the 5% of data that are too high or too low. This may lead to inaccurate parameters.

III. COMPARING TRAFFIC MODELS

In this section, we show the results obtained after comparing the models. The video traces used for the comparison are taken from [10] [11]. We used four traces (ARD talk, fitzek, Mobilkom and N3). These traces were used in many other performance evaluation research as in [8]. These trace were extracted from high-quality MPEG-4 videos having a duration of about 60 minutes and with various content (TV talk show, lecture, webcam). The videos were encoded using fixed GoP pattern IBBPBBPBBPBB. Other video traces of various types (MPEG-1, MPEG-2, MPEG-4 HD, etc.) were used to confirm the results mentioned in this paper but were not mentioned here for the sake of brevity.

For these models, the performance evaluation consists in testing how does the artificial traffic generated with each model is like the original trace. The artificial data feeds a network simulator that sends it through a number of routers with or without loss. The simulator measures the end-to-end delay, the jitter, the network load and the loss rate. For some models, we compare also quantile-quantile (Q-Q) plot for the original and the artificial traces. In all figures, we plotted confidence intervals but they can be too small to be visible in some of them.

For the simulation, the video frames are fragmented into IP packets with a maximum transfer unit of 1500 bytes. Each packet has a total overhead of 40 bytes for IP, UDP and RTP headers. We suppose that the transmission is not subject to errors. The packets are then forwarded into a network with five routers/servers in tandem. To evaluate delay, jitter and load, we supposed that the servers have infinite size buffers while for the evaluation of the loss rate, we used servers with finite buffer size. The servers processing time is proportional to the packet size and inversely proportional to link throughput. All links in the network have the same throughput.

The results depicted in Fig. 1 show that all models capture accurately the end-to-end delay while varying link throughput in the network. The accuracy is better for large values of throughput. This is explained by the queues in the

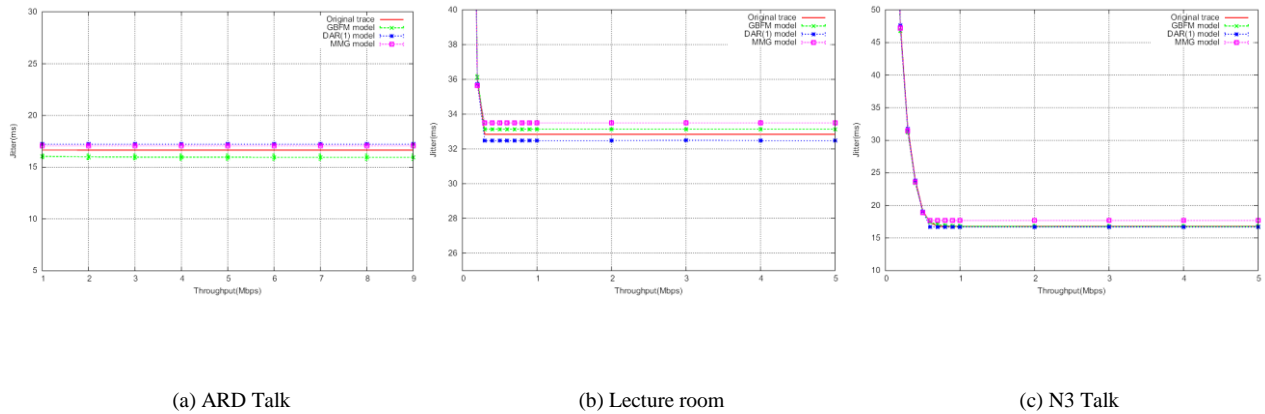


Fig. 2. Jitter for original and artificial traffic

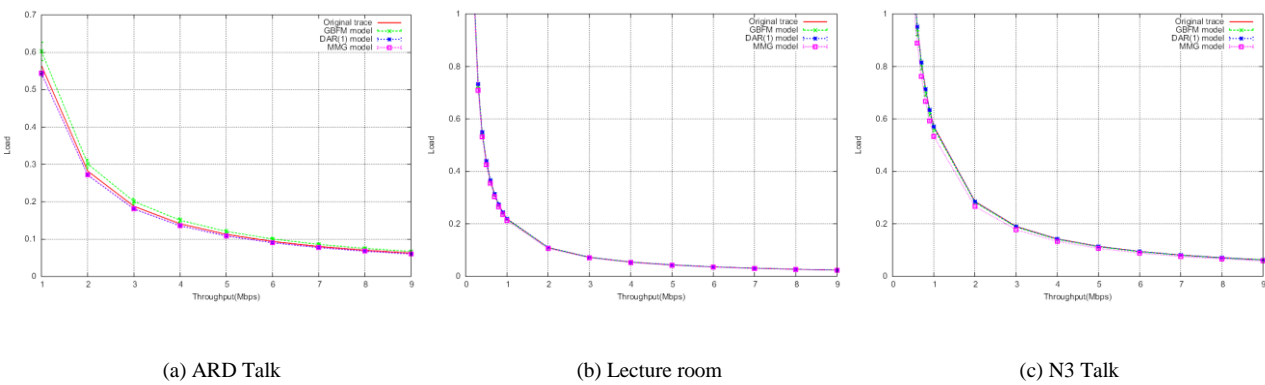


Fig. 3. Traffic intensity for original and artificial traffic

servers/routers that are emptied for large throughput which makes packets spending less time in the network and the transfer time is no longer dependent on other packets.

The figure shows also that our model GBFM is the closest to the original video trace. the DAR(1) model is less accurate. All three models under estimate delays for this network compared to original data. For all figures, we observe that GBFM generates random data with more scatter than other models. The standard deviation coefficient for GBFM is larger than other models. This explains in part the large size of confidence intervals in GBFM plots compared to other models.

A possible explanation of this fact is that in DAR(1) model data correlation is controlled by the model for each frame type I, P or B so that correlation in the artificial traffic and in the original trace is the same. As for MMG, a reason of its low standard deviation is that the MMG model selects a 99% percentile interval from the original data and does not consider extreme values in the trace. This leads to a more homogeneous set of data and smaller confidence intervals but may lead to a lower accuracy.

The Fig. 2 shows the variation of jitter. The values of jitter are obtained by computing the average inter-arrival time for all packets of the trace. In this simulation, we observe that jitter does not vary significantly when the traffic intensity is low (for high bitrate values). When traffic intensity is high,

we observe high peaks of jitter for all models. This fact can be associated to a lower video quality for the receiver. The average value of jitter is acceptable. We remark that both Markov models (GBFM and MMG) give accurate prediction of jitter. The GBFM model is the most accurate but it underestimates jitter values compared to trace and other models.

Traffic intensity is depicted in Fig. 3. The value of traffic intensity measures the amount of time the server is busy during the whole simulation. The figure shows that MMG model gives better results than other models. Our model is also accurate but its confidence intervals are larger than those of MMG model specially for small bitrates. For N3 talk video, the GBFM model outperforms other models. This observation leads us to note that generally, the model performance does not depend on a specific type of videos. Videos having the same type of content (tennis match, movie, cartoon or news for example) can give different performances when modeled with the same models. The effect of encoding has more effect than the type of content that is being encoded in the video.

To evaluate the loss rate, we changed the simulation scenario. the network is now reduced to a single hop and the source sends video rate to a server that has a limited buffer size. When packets reach the server and there is not enough space, the packets are rejected. We suppose that there are no Quality of Service (QoS) mechanisms to manage loss. We



suppose also that there is no other traffic in the network. The loss rate ratio is calculated by dividing the number of lost packets to the total number of packets sent in the network.

$\alpha$  and  $\beta$ . Though, the generation of random traffic is based on the Pearson distribution:

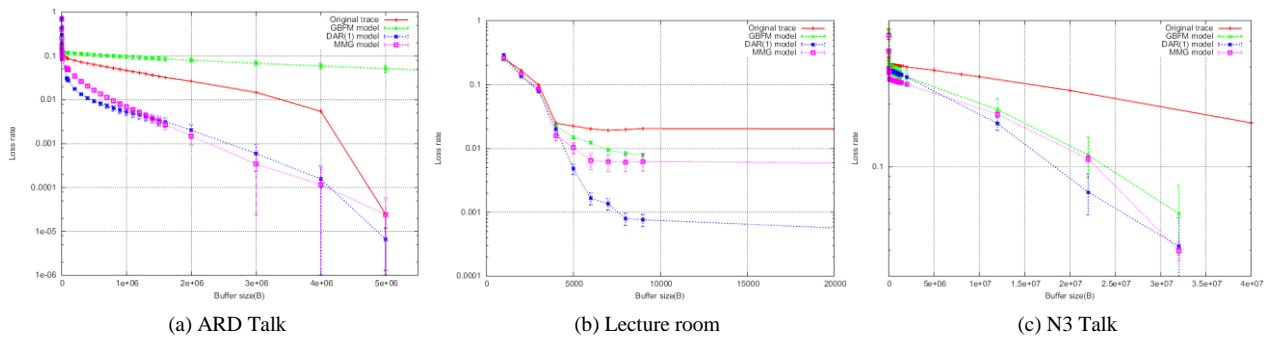


Fig. 4. Loss rate for original and artificial traffic

The Fig. 4 shows the loss rate for the original trace and the three models. Loss rate is plotted in log-scale to emphasize on the differences between models. These differences cannot be shown in normal scale because loss rates are small values. Results show that GBFM gives the most accurate results among the models. The DAR(1) model underestimates loss rate in most of the cases. Markov models give better results for low buffer sizes. The GBFM model trend is to overestimate loss rates comparing to MMG.

In Fig. 5, we plotted the quantile-quantile (QQ) information for artificial traffic of each model with the original trace. Each QQ plot is obtained by processing 100 artificial traces for each model and shows how close is the GoP size distribution to the original GoP trace which is represented by the reference line  $y = x$ . We observe that for the three videos, the curves are closer to the first bisector for small GoP sizes. All models fail relatively to capture well the highest percentiles of data. Nevertheless, we observe that GBFM model is the most accurate globally. The DAR(1) plot shows lower accuracy for intermediate to high percentiles.

For the lecture room video showing a webcam in an office (Fig. 5b), all models were not able to generate data in the highest 3% percentiles. These percentiles are located far from the other data points. Extreme high and extreme low percentiles show scattered data points and models trend is to generate lower frame sizes except for GBFM model in ARD video.

*Numerical considerations*

In order to compare the performance of the models, it is also important to consider numerical aspects to choose the most adequate model. We consider here space and time complexity and numerical stability of computation.

The DAR(1) model has the lowest time complexity among the models since it has a simple algorithm. It has also a small set of parameters: The auto-correlation parameter  $\rho$  and the Pearson type V probabilities with parameters

$$f(x) = \frac{x^{-(\alpha+1)} e^{-\beta/x}}{\beta^{-\alpha} \gamma(\alpha)}$$

The expression of  $f(x)$  is numerically unstable specially for high values of  $\alpha$  and  $\beta$  that are common in high quality traces. For example for webcam video, we have  $\alpha \approx 944.62$  and  $\beta \approx 8.58 \times 10^6$ . In such cases, the computation of the pdf generates an indeterminate form and requires specific computational attention to compute the random frame sizes. This may be a handicap if the model is to be implemented in an embedded system for admission control and QoS provisioning.

For the MMG model, it is necessary to create the GoPs, then the clips then the shots and the P matrix. Then, we need to compute the  $3(n + 1)$  parameters for each frame type of each shot and for the clip duration. For the artificial traffic generation, we generate gamma distributed random numbers. While the computational complexity is not very high, the number of parameters is relatively important.

For the GBFM model, after creating the GoPs, a classification algorithm is run to group GoPs into classes and to create scenes. The generation of the P matrix is the same than the MMG model but the creation of the phase-type distribution and the global Markov process is much more time-consuming. The generation of the random GoP sizes is based on random exponentially distributed sojourn time in each state and the use of Johnson distribution for the GoP size. The computation of frame size is obtained by simple divisions. The GBFM has a higher time complexity but requires fewer parameters (the A matrix and the drift vector D. In [9], the authors affirmed that the optimal results are obtained for  $n = 7$  while for the GBFM model, the optimal dimension of the A matrix is  $n = 3$  for a 2-state phase type processes [5].

IV. CONCLUSION

In this paper, we compared the performance of the fluid model with reference models and showed that while the three models capture well the trace statistics, the GBFM model is

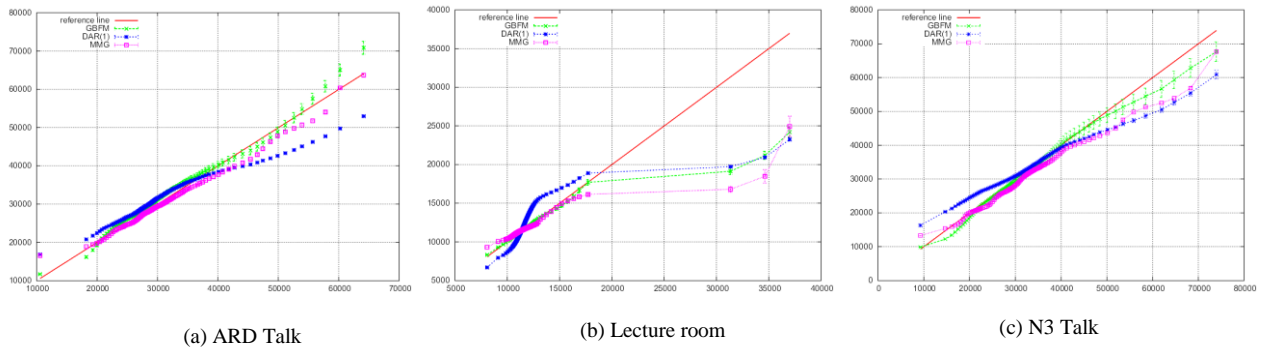


Fig. 5. Quantile-Quantile plots

the most accurate among them. DAR(1) model captures well auto-correlation between frame sizes but is not suitable for loss estimation. The GBFM has a large standard deviation compared to other models. The MMG model does not show such a dispersion because of 99% selection done in MMG. Quantile-quantile plots shows that most model fail to capture accurately the statistical characteristics of the highest percentiles of frame sizes.

While DAR(1) model captures well frame size statistics according to [8], the analysis of GoP size distribution shows a lower performance because frames of different types in the same GoP are modeled independently and do not reflect the fact that P and B frames are generated using I frames. For MMG and GBFM models, the model focuses on the generation of frame sizes according to their context in the video (scenes or clips) but the generation of intra-GoP information is based on simple linear models.

As a conclusion, the results show that there is no model that outperforms the others clearly but we can say objectively that GBFM is the most accurate without a significant additional complexity.

## REFERENCES

- [1] Cisco Systems, "The Zettabyte Era: Trends and Analysis, white paper, May (2015)
- [2] Hlavacs, H., Kotsis, G., Steinkellner, C., "Traffic source modeling," Tech. rep., Institute for Appl. Comp. Science and Inf. Systems, University of Vienna (1999).
- [3] Izquierdo, M., Reeves, D., "A survey of statistical source models for variable-bit-rate compressed video," *Multimedia Syst.* 7(3), 199-213 (1999).
- [4] Avramova1, Z., Vleeschauwer, D.D., Laevens, K., Wittevrongel, S., Bruneel, H., "Modelling H.264/AVC VBR video traffic: Comparison of a Markov and a self-similar source model," *Telecommunication Systems* 39(2), 91-102 (2008).
- [5] Abbessi, W., Nabli, H., "GoP-based fluid markovian modelling of video traffic," *International Conference on Communications and Networking (ComNet 2010)*, pp. 1-8, Tozeur (2010).
- [6] Tanwir, S., Perros, H., Anjum, B., "A QoS evaluation of video traffic models for H.264 AVC video," *5th International Conference on Next Generation Networks and Services (NGNS)*, Casablanca (2014).
- [7] Likas, A., Vlassis, N., Verbeek, J., "The global k-means clustering algorithm," *Pattern recognition* 36(2), 451-461 (2003).
- [8] Lazaris, A., Koutsakis, P., "Modeling multiplexed traffic from H.264/AVC videoconference streams," *Computer Communications*, vol. 33, no. 10, pp. 1235-1242, 2010.
- [9] Sarkar, U.K., Ramakrishnan, S., Sarkar, D., "Modeling full-length video using markov-modulated gamma-based framework," *IEEE/ACM Trans. Netw.*, 11(4), pp. 638-649, 2003.
- [10] Fitzek, F., Reisslein, M., "MPEG-4 and H.263 Video Traces for Network Performance Evaluation," *IEEE Network*, Vol. 15, No. 6, pages 40-54, November/December 2001.
- [11] <http://www-tnk.ee.tu-berlin.de/research/trace/trace.html>