

Estimating Video Quality over ADSL2+ under Impulsive Line Disturbance

Glauco Gonçalves¹, Ramide Dantas¹, André Palhares¹, Judith Kelner¹,
Joseane Fidalgo¹, Djamel Sadok¹, Henrik Almeida², Miguel Berg²,
and Daniel Cederholm²

¹ Universidade Federal de Pernambuco, Brazil

{glauco, ramide, andre.vitor, jk, joseane, jamel}@gpert.ufpe.br

² Ericsson Research, Sweden

{henrik.almeida, miguel.berg, daniel.cederholm}@ericsson.com

Abstract. QoS Provisioning for 3P over xDSL remains a challenging task due to the effects of line impairments on such services. Differently from simple data, video and voice services have strict requirements for loss and delay tolerance. The accurate assessment of final service quality is part of this provisioning process, but its direct measurement is yet not practical. In this paper we explore the possibility of estimating service quality, with focus on video delivery, by investigating its relationship with performance data available to xDSL operators and deriving models for estimating quality from this data. Experiments using a real xDSL platform and different noise types were conducted. The derived models showed to be accurate enough to estimate video quality for the scenarios evaluated.

Keywords: Video, QoE, ADSL, performance estimation, noise.

1 Introduction

The provision of triple play (voice, video and data) services over ADSL/2/2+ technology for residential and business customers remains a challenging task when considering quality assurance. Parasitic effects on the physical layer (L1) of the access network like non-transient and transient noise, up to now negligible for the pure best-effort Internet service, are starting to play a damaging role when it comes to the transport of real-time video content. To meet the quality requirements for the transportation of these new applications over the network, an end-to-end, real time quality monitoring architecture must be part of the access infrastructure. Nonetheless, such architecture remains a long solution ahead for research to pursue.

Existing research into triple play delivery often assumes simple models for packet loss and delay using well-known statistical distributions. The reality faced on the field is much more complex than this simplified view. The end-loop is subject to different types of noise effects and physical impairments that have shown to be hard to capture and model [12]. In this paper we take a different path for estimating the Quality of Experience (QoE) in that we study practical scenarios submitted to varying noise

levels. We show special concern with impulse noise, seen as one of the most harmful noise types, in order to discover and model its impact on video delivery.

Service providers often do not have online access to user feedback nor can they perform deep packet inspection or similar traffic analysis to assess users' QoE. Given the limited applicability of such methods, we opted for a different approach. We investigated how line data provided by DSL equipments is correlated to video streaming quality, thus making it possible to build useful models for video QoE estimation. Operators can embed such models into tools that proactively monitor and adapt line settings for changing scenarios. By doing this, operators can preserve service quality and fulfill customer expectations.

QoE may be seen as a cross cutting concern, depending on physical, network and application layer performance, and therefore we consider metrics on all these layers in our investigation. At the application level, the focus was on the Peak Signal-to-Noise Ratio (PSNR) metric, a well-deployed metric to measure video quality in an objective way [1], [2]. Note that the authors do not claim that this is the most important metric nor that it tells the whole story on its own. At the network level we measured packet loss ratio, which is known to affect video quality strongly [6]. At the physical layer we have collected DSL metrics [10] such as the number of damaged blocks received at a user's modem, line bit rate, actual INP (Impulse Noise Protection), and actual SNR (Signal-to-Noise Ratio) margin, among others. By finding the correlation between those metrics we were able to find models for estimating application and network metrics from physical ones, bridging the gap between these otherwise disjoint figures.

In order to investigate ADSL and video metrics' correlation, a series of experiments were carried out using diverse line settings and environment conditions, such as loop lengths and noise patterns. The data obtained from these experiments was used to feed statistical and mathematical tools to calculate the dependence among metrics and derive models for high-level metric estimation. The models obtained were checked against validation data to verify their accuracy.

The experiments were performed using a test-bed deployment of a commercial ADSL2/2+ platform and performance data was obtained from actual measurements on this platform. We focused on evaluating SDTV-quality (Standard Television), MPEG-2 video streaming over ADSL2+. An impulsive noise pattern was injected in the ADSL line during video streaming and performance data was collected at both sides of transmission and from DSL equipments.

The rest of this paper is organized as follows. Section 2 describes the experiments setup, including the noise generator we used to inject impulsive noise in the experiments. Section 3 analyses the results obtained from these experiments. The regression models for video quality estimation as well as the methodology used to derive them are presented in Section 4. Section 5 presents some related work and Section 6 draws final remarks and topics for future works.

2 Experiment Configuration

A series of measurements were made in order to acquire performance data of video over ADSL2+. Such experiments were performed using a controlled ADSL2+

network, comprised of equipments such as a DSLAM and a CPE modem, equipments for line emulation and noise generation as well as for video streaming and capture. The experiment procedure is detailed in section 2.1, section 2.2 presents the model used for noise generation, and section 2.3 shows the configuration parameters used.

2.1 Testbed and Experiment Procedure

The ADSL testbed used for the measurements is outlined in Fig. 1. For streaming video content, two Linux boxes were used at the extremes of the DSL line. One behaved as a multimedia client, which receives the video transmitted by the multimedia server. The client is connected to the DSL line through an external DSL modem while the multimedia server is connected to the DSLAM. The VideoLAN Client (VLC) [5] media software was used to stream the video content from server to client sides using UDP as the underlying transport protocol.

The line emulator equipment provided the physical media between the customer premises (CP side) and the central office (CO side), where the DSLAM is placed. An arbitrary wavelength generator (AWG) and a noise injection unit were used in conjunction with the line emulator, allowing various line environments to be tested. While the line emulator provided a wide range of loop length possibilities, the AWG was used to generate different noise patterns into the line.

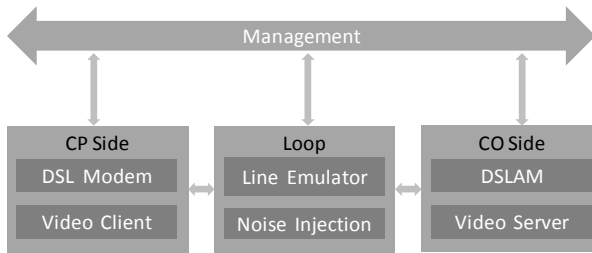


Fig. 1. Test bed scheme

The experiment procedure involves four main steps: 1) configure experiment parameters; 2) activate DSL line and wait for modems synchronization; 3) stream the video from the CO to the CPE side (downstream); and 4) collect the interest metrics at endpoints and DSL equipments. Noise was injected at the CPE side before line synchronization. Therefore, actual line settings could not match exactly configured parameters since the modems try to achieve a better protection level given the line conditions. Injecting noise before synchronization has the advantage of providing more stable experiments when compared to post-synchronization noise injection.

ADSL-related metrics were collected directly from the DSLAM via SNMP. The main metrics collected are presented in Table 1. (more details on these metrics can be found in [10]). Note that these metrics are related to the downstream DSL channel, since the video traffic flows only from the CO side to the CP side. We extracted network-related metrics as packet loss, delay, and jitter from video traffic traces captured at each side of the ADSL line. Moreover, PSNR was calculated afterwards

Table 1. Metrics Collected for Layers 1 and 2

Metric	Description
CRC	Uncorrected FEC Blocks received at CPE
FEC	Corrected FEC Blocks received at CPE
ES	Errored Seconds
SES	Severely Errored Seconds
UAS	Unavailable Seconds
Rate	Synchronized Line Rate
ID _{act}	Actual Channel Interleave Delay
INP _{act}	Actual Impulse Noise Protection
SNR _{act}	Actual SNR margin

using the original encoded video and the transmitted video in order to measure the quality of the video received by the client.

MPEG-2 video streams were used in the experiments. Streams were encoded at bit rate of 4Mbps and 30 frames per second, with image size of 704x480. This choice was made with the goal of characterizing SDTV (Standard Definition Television) video transmission. Two video streams were used: one with 15 seconds of duration and another with about 1 minute. The first video was used to create the regression models and is referred to as “tennis video”. The second one was used for validation purpose and is referred to as “bridge video”.

2.2 Noise Modeling and Generation

Several types of noises can affect DSL systems, being the Repetitive Electrical Impulse Noise (REIN) one of the most severe of them [12]. REIN is commonly found on the CP side, being its main sources badly shielded household appliances, illumination devices and switching power supplies used in PCs [15]. REIN was chosen for this evaluation given both its severity and common occurrence in DSL installations. REIN is also simpler to model and generate given its more predictable nature compared to random Impulsive Noise. In our study, we generated REIN using an arbitrary wavelength generator based in the model summarized below. We describe the modeling parameters that were used in the experiments in order to shed some light on the physical effect of each model parameter.

A REIN signal $x(t)$ is described as a periodic sequence of bursts as shown in Fig. 2.. The bursts’ temporal spacing is denoted by T and defines the periodicity of the noise signal. A burst $x_B(t)$ itself consists of a sequence of N_B base signals $x_S(t)$ with duration T_R . The duration of a burst is denoted by T_B and is clearly given by $T_B = N_B T_R$. The base signal is a sized version of a normalized peak-peak noise shape function $g(t)$ with support $-T_R/2$ to $+T_R/2$. The REIN signal is offset in time by T_0 .

The noise shape function $g(t)$ is defined in the time-interval $|t| \leq T_R/2$ with peak-peak value normalized to 1. It can take different forms depending on the desired frequency content. For our experiment we used the sinc function ($\text{sin}(t)/t$).

When dealing with REIN generation, we focus on these four parameters: 1) the periodicity of the bursts, controlled by the parameter $f = 1/T$; 2) the number of base signals per bursts N_B ; 3) the periodicity of the base signal inside a burst, which is

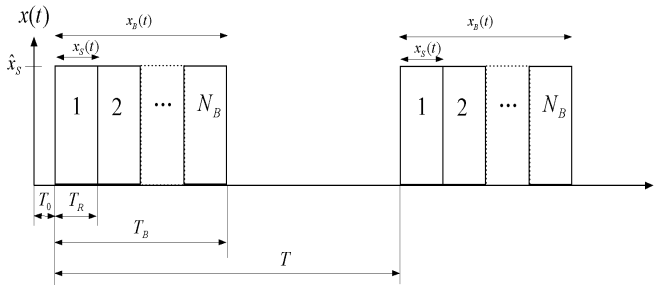


Fig. 2. REIN Signal Composition

controlled by $f_R = 1/T_R$; and 4) the power of the burst, which is given in dBm for 50 Ohms impedance.

2.3 Configuration Parameters

The main problem to deal with real experimentation is the great number of parameters that must be carefully configured to guarantee interesting results. Moreover, parameters configuration has a tradeoff between representative scenarios and viability of experiments in terms of execution time. Considering this tradeoff we varied noise profiles, line protection settings, and loop lengths. The ones we have fixed were noise power level and DSL parameters like maximum interleave delay. Such choice allowed the verification of the video quality under different environment impairments and the effectiveness of the protection mechanisms commonly used by operators.

Table 2. Noise profiles configurations

Profile	Burst Freq. (Hz)	Burst Length (Pulses)	Burst Length (us)	Comments
0Z	-	-	-	No noise
3X	100	25	100	Aggressive
3Y	100	250	1000	Very aggressive

Using the REIN model described in previous section, we can characterize noise by its burst frequency and the number of spikes in each burst (or burst length). Combining these parameters we defined three noise profiles used in the experiments, shown in Table 2. The 0Z profile indicates an environment without noise that is used for reference, while 3X and 3Y indicate aggressive noise profiles. While the former affects one DMT (Discrete Multi-Tone) symbol certainly and can affect up to two consecutive DMT symbols, the latter corrupts four consecutive DMT symbols completely and can affect partially up to 2 additional symbols. Noise power levels of 3X and 3Y profiles were fixed to -24.71 dBm (13 mV), which is compatible with the power range of the impulse noise model presented in [12]. This level determines the amplitude of individual peaks inside each noise burst and does not represent the noise average power, since it depends on burst length and its frequency.

Line protection settings, comprising here Impulse Noise Protection (INP) and SNR margin (SNR_{mar}), were also varied. Values for those parameters are not directly set but provided in terms of a minimum for the INP (INP_{min}) and a target value for the margin ($\text{SNR}_{\text{mar,tar}}$)¹. The modems try to configure the line during synchronization so that those constraints are respected. After line synchronization, the line presents actual INP (INP_{act}) and margin ($\text{SNR}_{\text{mar,act}}$) values which may differ from the configured ones (INP_{min} and $\text{SNR}_{\text{mar,tar}}$).

SNR margin and INP deal with noise in different ways. Higher SNR margin values prevent the transmitted signal of being corrupted. If such protection is not effective, corrupted data can be recovered using redundancy data sent along with user data. The effectiveness of redundancy can be further improved if various data frames are interleaved, hence spreading the effects of noise bursts across data frames and allowing for better correction. Redundancy and interleaving features are controlled via the INP_{min} parameter. Formula (1) below gives the theoretical definition of INP for a DMT symbol of L bits, FEC (Forward Error Correction) frames of N bytes, being R bytes of redundancy, and D the frame interleave depth [11].

$$\text{INP} = D \times \frac{R}{2} \times \frac{1}{L/8} \quad (1)$$

The SNR margin provides protection against the noise by trying to ensure a signal-to-noise ratio that keeps the BER (bit error rate) below 10^{-7} , what can ultimately decrease the achieved rate. The SNR margin values used in the experiment were 6 dB (a typical value), 12 dB and 18 dB. The values used for INP_{min} were 0, 2 and 4 DMT symbols, which were found to be more applicable in practice. Table 3. Summarizes the experiment parameters.

Table 3. Experiments parameters and values

Parameter	Values
Target SNR margin (dB)	6, 12, 18
Minimum INP (DMT symbols)	0, 2, 4
Noise Profiles	0Z, 3X, 3Y
Loop lengths (m)	1000, 2000, 3000

Each experiment was repeated 10 times providing then 10 samples for each combination of parameters evaluated. This number of samples was determined by previous experiments and provided a good trade-off between statistical quality of the measurements and the time demanded to perform them.

3 Experiment Results Analysis

The graph in Fig. 3 shows the average percentage of lost packets for each experiment configuration. The x-axis presents the combinations of values for noise profile, INP_{min}

¹ Minimum ($\text{SNR}_{\text{mar,min}}$) and maximum ($\text{SNR}_{\text{mar,max}}$) values for the margin need also to be provided. For all experiments, $\text{SNR}_{\text{mar,min}} = 0.9 * \text{SNR}_{\text{mar,tar}}$ and $\text{SNR}_{\text{mar,max}} = 1.1 * \text{SNR}_{\text{mar,tar}}$.

and $SNR_{mar,tar}$ used in each experiment (10 replications per configuration were made; average loss for each replication is plotted). Each loop length value is represented with a different mark. As expected, in general, more packet loss occurred for low protection scenarios, especially for $INP_{min} = 0$ and $SNR_{mar,tar} = 6$ dB. Loss reached 80% under the most aggressive noise (3Y) and the longer loop (3000m).

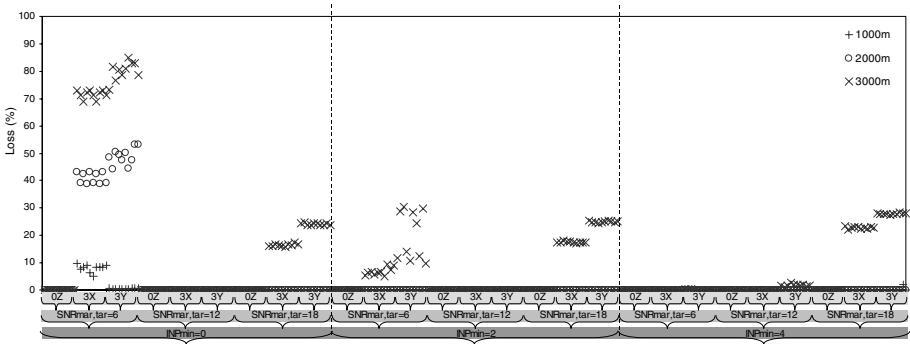


Fig. 3. Packet loss ratio for diverse line and noise configurations grouped by loop length. Each sample represents the average loss ratio over each experiment round.

With higher protection, that is, $INP_{min} = 2$ or 4, no losses were detected for loops of 1000 m and 2000 m. But, for such INP settings, losses between 20% and 30% were found with the 3000 m loop, mostly caused by lack of bit rate, except for the singular case of $INP_{min} = 2$ and $SNR_{mar,tar} = 6$ dB where losses are caused by noise. This lack of bit rate occurs when noise is present during line synchronization and the higher actual INP pushed the line rate lower than the video streaming needs. Higher $SNR_{mar,tar}$ (18 dB) also contributed to decrease the rate and cause more losses. The available bit rate under $SNR_{mar,tar} = 18$ with noise profile 3Y for different loops and INP_{min} values is shown in Table 4. Notice that under these conditions the DSL line rate is below video rate needs for the 3000 m loop, resulting in ordinary packet loss.

The best scenario regarding packet loss was the one with SNR_{margin} set to 12 dB for any INP setting. This configuration provided almost no losses, with the exception of the 3000m loop, 3Y noise, $INP_{min}=4$ case, where losses were lower than 5% and were caused by lack of bit rate.

With respect to the PSNR, the results were mostly influenced by packet loss, with lower PSNR values occurring when losses were detected. The 3000m loop was more affected since it suffered higher loss ratios. An important aspect of PSNR is that little

Table 4. Average line rate in Mbps for diverse INP_{min} and $SNR_{mar,tar} = 18$ dB (3Y noise profile)

INP_{min}	Loop length		
	1000 m	2000 m	3000 m
0	17.57	9.10	3.62
2	15.27	8.75	3.57
4	12.19	7.96	3.46

losses were enough to cause major damage to video quality. Values near 15 dB of PSNR were observed when packet loss exceeded only 3% on noisy and low bit rate scenarios, indicating very low video quality. For lower losses, the PSNR was near or above 30 dB, which means good or very good video. These results were decisive to our modeling as explained in the next sections.

Measurements of delay and jitter had been taken but are not shown in this paper, since their impact on video quality was negligible. However, it must be noticed that this is valid only for the evaluated scenarios. Our experiments used CBR video traffic and no background traffic. Bursty background traffic as Web applications or variable bit rate videos can cause peaks of congestion at the DSL line and would affect the video quality sensibly. Quality models considering delay and delay variation will be approached in future works.

3.1 Metric Correlation

In this section the Pearson's correlation between the most relevant metrics investigated is presented and analyzed. The correlation table with these metrics is shown in Table 5. This table is based on the tennis video data from all the different loops. Some values are highlighted for fast lookup during the explanations.

Initially the correlation coefficient was computed for all data without any data segmentation as presented in the last column of Table 5. The obtained coefficients showed low correlation between metrics, most of them very below 0.9. To improve these results, the data was divided using our previous knowledge on the behavior of video transmission over a noisy ADSL line, as will be explained in Section 4. Right now, it is important to show that with the data segmentation the correlation coefficient increased significantly. For example, the CRC metric that would be the main metric for packet loss correlation showed low correlation considering all data. This occurs because in noisy scenarios with low protection the management data flowing upstream containing the CRC value is corrupted by the noise while querying the DSLAM modem². The segmentation increased CRC correlation for high protection scenarios. Moreover, segmentation increased the correlation coefficient for metrics that compute errored seconds such as ES, SES and UAS.

Table 6 presents the correlation between PSNR and some physical metrics and network metrics. As expected, correlation between packet loss and PSNR is high and inverse. On the one hand, with high losses during transmission, the quality of a video will be affected negatively and the PSNR decrease. On the other hand, transmissions with no loss will result in high PSNR values. Additionally, the correlation between PSNR and packet loss was more significant than the other correlations tested.

An approach for PSNR estimation would be by creating a model directly between PSNR and low-level metrics. However, we opted to estimate packet loss and then use this estimation to obtain a qualitative PSNR model. The main factor behind this choice was our previous knowledge of the behavior of the loss relatively to lower level metrics. Also, it is known that there is a non-linear relation between packet loss and PSNR [6], a result that was in fact verified in our experiments.

² In these cases, the CRC value retrieved is zero.

Table 5. Correlation coefficients for packet loss ratio and physical metrics for all data and divided in each category

Metric	Packet Loss			All
	Rate < 5 Mbps	Rate \geq 5 Mbps		
		INP _{act} <2	INP _{act} \geq 2	
CRC	-*	0.330	0.959	0.322
Rate	-0.998	-0.308	-0.196	-0.254
INP _{act}	-0.186	0.326	-0.173	-0.171
ES	-	0.662	0.795	0.562
SES	-	0.849	0.856	0.671
CRC ²	-	0.328	0.993	0.335
ES*INP _{act}	-	0.948	0.688	0.367
SES*INP _{act}	-	0.978	0.837	0.457

*The correlation could not be calculated since metrics presented no variation (metrics equal zero in these cases).

Table 6. Correlation coefficients for PSNR and physical metrics for all data

Metric	PSNR
CRC	-0.4289
Rate	0.2997
INP _{act}	0.1177
ES	-0.6696
SES	-0.6356
CRC2	-0.3572
ES*INP _{act}	-0.5053
SES*INP _{act}	-0.4913
Loss %	-0.7381

4 Video Performance Estimation Models

4.1 Modeling Methodology

Our strategy was to keep models as simple as possible while achieving acceptable accuracy, i.e. coefficient of determination equal or above 0.9. By simple models we mean models relying on only a few variables (or combinations of them). The generated models through the application of regression techniques on all available data presented low coefficients of determination, showing that such simplistic approach should be avoided. Then, we decided to segment data and group variables considering our previous knowledge on DSL and video performance.

The entire data set was separated based on one or more variables to which were applied thresholds, as will be detailed further ahead. With the segmented data, the accuracy of the models increased significantly. For example, data was segmented based on the DSL line rate to separate scenarios where the loss was caused by lack of bit rate of those caused exclusively by noise.

Some predictor variables with similar effects on the response variables were grouped in categories based on this similarity on their semantics. For example, ES, UAS and SES, which all stand for errored seconds were placed together in the same group. It was selected the variable with the greater correlation value in that group, in other words, the one with strongest influence in the response. Some of the variables were combined (by multiplying them) following insights obtained looking at results behavior.

With data segmented and the input variables selected, the next step was to generate the models. At first, models were derived using the multiple linear regression technique and, when a non-linear relation was clearly visible on scatter plots, polynomial regression method was employed. Coefficients whose confidence intervals made them crosscut zero were discarded to simplify the models.

Finally, we validated the models by evaluating the model obtained from the tennis video data with data acquired from the validation video (bridge). At a first moment, the models for each data segment were validated separately. That was reasonable because, when a particular model does not show a representative behavior, it could be fixed separately. In a second moment, we joined all the models estimates and compared them directly with all the validation data. Thus, we could see that the general model is satisfactory.

4.2 Estimating Packet Loss

To build a suitable packet loss model, data was segmented in three categories according to the packet loss behavior. In each category we discovered interesting correlations between the loss and specific metrics or metric combinations. Table 5 shows the most relevant correlations which were used to generate the models for each scenario.

In a first analysis of the data, it was noticed that the data could be divided using the DSL synchronized line rate, since when the line rate is below the needed video bit rate packet loss will occur due to lack of bit rate. Since the tested video bit rate was 4 Mbps, the data was divided into two initial categories at the rate of 5 Mbps. This value was chosen considering header overhead and giving some bit rate for other traffic types in the channel including possible background traffic.

When the rate is lower than 5 Mbps, we observed that the packet loss was strongly correlated with the rate (as highlighted in Table 5.). We generated a linear model using the rate as the explanatory variable. The model obtained is numerically given by the formula (2), where *loss* is the packet loss ratio (ranging between 0 and 1) and *rate* is given in Mbps.

$$loss = -0.2 \times rate + 0.97 \quad (2)$$

When rate is greater than 5 Mbps we observed that the CRC metric presents high correlation when the actual INP (INP_{act}) is greater than or equal to 2, since in this situation line is protected and thus the chance of error in the measured CRC is small. For the other case, when actual INP is less than 2, the SES and ES metrics presented high correlation values. When the INP_{act} is greater than 2, the model obtained is given by:

$$loss = 0.003 \times crc^2 \tag{3}$$

In formula (3), *crc* is given in thousands of CRC events, which represents the number of corrupted blocks computed at the client side. Please note the non-linear (quadratic) correlation between CRC and packet loss. This model was obtained using polynomial regression with degree 2. As the lower degree coefficients were insignificant, they were removed from the final model.

When INP_{act} is less than 2, we generated a model based on SES, ES, and the actual INP. We observed the following relation:

$$loss \propto (\alpha SES + \beta ES) \times (1 + \gamma INP_{act}) \tag{4}$$

Coefficients α and β in formula (4) are weights of each parameter in the weighted sum between SES and ES. The multiplication by INP_{act} indicates that the damage observed by SES and ES metrics are associated with protection employed, where higher INP values indicate more data loss. One is added to INP_{act} to avoid the relation becoming null when INP is zero. The model obtained for this category is given by:

$$loss = a \times SES + b \times ES + c \times SES \times INP_{act} + d \times ES \times INP_{act} \tag{5}$$

Where $a = 8.1 \times 10^{-4}$; $b = 5.9 \times 10^{-5}$; $c = 9.7 \times 10^{-3}$; $d = -4.8 \times 10^{-3}$.

The weights associated to SES and ES reflect the fact that SES events represent a more harmful condition than ES events, i.e. more packet losses occurred when SES is observed since more CRC events are necessary to trigger a single SES event. Multiplying SES and ES by the actual INP means that, since errors occur (i.e. the protection employed was not effective), higher INP values indicate higher losses. One possible explanation is the fact that higher INP implies deeper interleaving, and since the protection was not able to prevent data corruption, the error tends to be spread across more disperse FEC frames, affecting more packets and leading to a reverse effect than the one expected from the usage of INP.

After generating the models, we evaluated and validated them. Table 7 shows the coefficient of determination obtained for each category in both scenarios: modeling and validation. The table also presents the coefficient of determination for the general model.

The general model was obtained combining all sub-models and, despite the increased complexity by use of three models, the general model implementation does not require more complex operations than comparison and basic arithmetic. Fig. 4 presents the scatter plot between the estimated packet loss and the measured loss for modeling and validation data. These plots show that, despite some imprecision, the generated model is a good approximation for both data sets.

Table 7. Coefficients of determination for each category

Category	Tennis Video (Modeling)	Bridge Video (Validation)
Rate < 5 Mbps	0.997	0.995
Rate ≥ 5 Mbps and $INP_{act} < 2$	0.983	0.962
Rate ≥ 5 Mbps and $INP_{act} \geq 2$	0.987	0.831
General Model	0.987	0.968

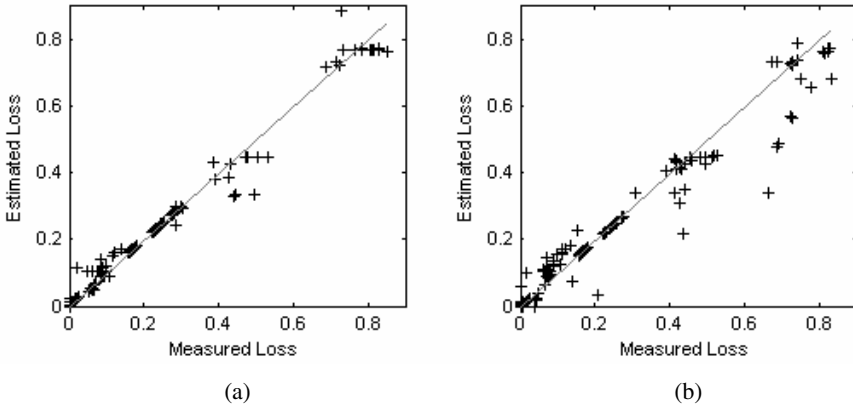


Fig. 4. Scatter plot of measured and estimated loss: (a) modeling data (b) validation data

4.3 Estimating Video Quality

To model the PSNR we chose to use the packet loss, supported by the known relation between these metrics. In [6], it is shown that network layer metrics, as packet loss, have direct impact on application layer metrics such as PSNR. Further, [6] mentions that video quality and packet loss present a non-linear relation. Our results demonstrated this non-linear relationship as showed in Fig. 5. This was the case for both real (measured) and estimated packet loss, where estimated values were generated using the models described previously.

Based on this non-linear behavior, we tried to fit several non-linear functions on the data, but, these fits showed low accuracy. The reasons for this undesirable behavior of the PSNR metric are twofold. First, PSNR does not have a standard upper limit: the best value is achieved when the difference between the received video and the original video is zero leading PSNR to tend to infinite. To avoid this, a 100 dB bound was defined as an arbitrary upper bound. Second, PSNR behaves as a categorical variable.

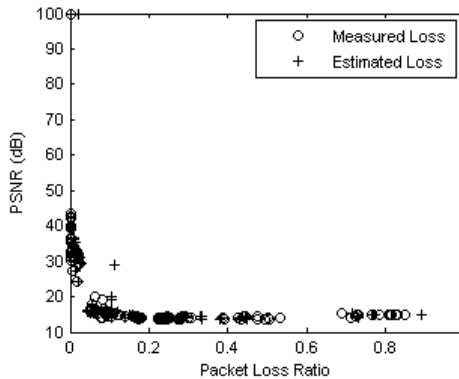


Fig. 5. Scatter plot between the measured loss and the measured PSNR

It is possible to map PSNR values into MOS-based categories, reproducing the human visual perception. Thus, following the relationship between PSNR and MOS presented by [4] we created three categories showed in Table 8.

Table 8. PSNR to Video Quality Mapping

PSNR (dB)	Quality
> 40	Excellent
> 30 and ≤ 40	Good
≤ 30	Poor

We applied these categories to our data to infer thresholds based on the packet loss, obtaining a simple categorical model. By observing the scatter plot between the measured packet loss and PSNR, a packet loss ratio of up to 3% was found to designate *Good* video quality while the threshold of 1% represents *Excellent* video quality. These thresholds revealed accuracy of about 99% in the *Good* video quality and about 98% in the *Excellent* video quality for the modeling video data. This accuracy is calculated as the percentage of correct predictions over the total number of samples.

In practice it is necessary to work with loss prediction since the actual packet loss is not available. Thus, we employed the thresholds using the estimated packet loss. We also verified these thresholds using validation data. All accuracy results were satisfactory as can be verified in Table 9.

Table 9. Accuracy for different scenarios

Data	Video Quality	
	Excellent (Loss < 1%)	Good (Loss < 3%)
Tennis (Measured Loss)	97.8%	99.3%
Tennis (Estimated Loss)	99.3%	99.4%
Bridge (Measured Loss)	100%	100%
Bridge (Estimated Loss)	97.1%	98.7%

Given these good results, we can formalize the model as shown in formula (6), where *loss* is the estimated percentage of lost packets.

$$\text{VideoQuality} = \begin{cases} \text{Excellent} & \text{if } (\text{loss} < 1\%) \\ \text{Good} & \text{if } (1\% \leq \text{loss} < 3\%) \\ \text{Poor} & \text{if } (\text{loss} \geq 3\%) \end{cases} \tag{6}$$

Fig. 6 contains frames taken from the tennis video for different loss conditions, allowing the visualization of the different quality levels. Fig. 6(a) shows the original frame for reference. At Fig. 6(b) we can see the same frame with “excellent quality”, as the thresholds in Table 8, whose PSNR is 40.5 dB. Little distortion is perceived,

confirming the category. Fig. 6(c) shows a frame with PSNR = 32 dB, meaning a “good quality” video. In this frame we can see some distortion caused by packet loss, although the scene can still be understood. It must be noticed that the PSNR of this frame approaches the lower bound threshold of 30 dB for “poor quality” videos. With PSNR = 21 dB, the video showed in Fig. 6(d) is below this lower bound: its scene cannot be understood properly, although the scene context is still preserved. This phenomenon is caused by the temporal compression employed by MPEG-2 encoders. When there was movement in the scene, packet loss causes loss of information and consequent distortion; scene background was reutilized from previous frames and thus preserved. This last case falls into the “poor video quality” category.

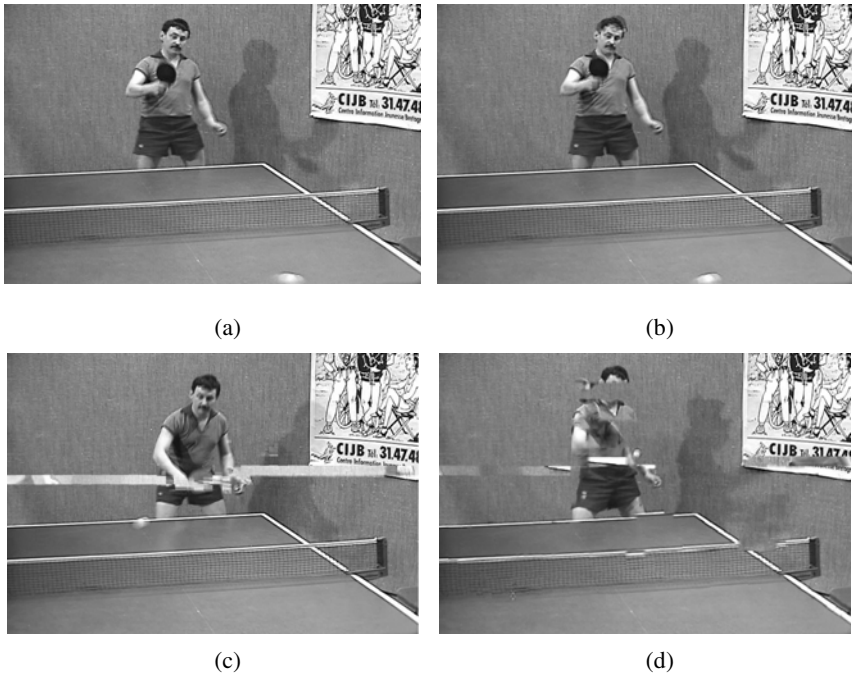


Fig. 6. Sample frames for different video qualities: original video (a); excellent quality video (b), PSNR=40.5dB; good quality video (c), PSNR=32dB; poor quality video (d), PSNR=21dB

5 Related Work

Various studies of real-time QoE estimation for video applications can be found in the literature. The study developed in [7] uses a simulated network with real video traces to evaluate the impact of packet loss, packet error (caused by noise), delay and jitter on application level quality metrics such as PSNR. The authors conclude that the packet loss is the most degrading event for video quality, but does not discuss how to estimate the video quality based on packet loss as we have done in this work.

In [13], authors develop linear models to estimate a MOS-like subjective quality metric for audio and video transmissions based on application metrics like audio/video

synchronization and MSE as well as the video content. Their experiments were run over an Ethernet network with web traffic generating disturbances on video traffic. Results had shown that the proposed approach can estimate video quality with high accuracy. The main difference to our approach is that they used application metrics instead of low-level metrics as we have used.

In [14] was derived analytically a relative PSNR metric, which is a difference between the actual PSNR and a reference PSNR. Besides packet loss effects, the metric considers impact of codec selection, the packetization scheme, and the video content. As pointed by authors, their quality metric underestimates impairments caused by bursty loss events, making it not suitable to environments subject to impulsive noise as xDSL networks.

The studies developed in [8] and [9] are similar to our own. In the former, using a non-reference video quality metric, the authors applied linear regression to estimate the video quality based on layer 3 metrics over a simple emulated network, differently from our environment, which uses real equipment and focuses on the ADSL access network. The latter study modeled the subjective MOS [3] for an interactive game application, based on measurements taken for a real gaming network. The obtained model, interestingly, differently from us, does not consider packet loss, since the experiments conducted by the authors showed that packet losses up to 40% have little impact on their gaming application. Video applications cannot make such assumption, as they are very sensitive to packet losses.

6 Conclusions

This paper presented models for estimating video quality of experience under several noise and line configuration scenarios. It is worth noticing that, unlike most existing works in the literature, our models were built upon experiments on a real DSL test-bed. Our experiments were by no means exhaustive but we have been able to obtain representative models and important results that we would like to discuss with the community to build further on them.

From a more practical perspective, we believe that the models derived in this work could be directly integrated into a real-time performance monitoring tool to evaluate per-user QoE in an operational xDSL plant. The procedure used to derive such information can be extended, by employing IA techniques or feedback-based approaches, to allow an automatic adjustment of the weights of model variables, allowing for better adequacy to different scenarios.

We are currently working on models for the estimation of other network metrics and their correlation with video quality under noisy conditions. More sophisticated scenarios, including varying background traffic, are going to be investigated.

Furthermore, new experiments can be done by expanding the validation area of our models including the use of other video codecs (e.g. MPEG4). Our goal in this case is to broaden the applicability of our models, as we realize that performing all the possible tests is an unfeasible task. Consequently, building tools for the correct estimation of channel quality especially under noisy conditions is a difficult task, and any real step towards this is encouraging and extremely helpful.

References

1. Fitzek, F., Seeling, P., Reisslein, M.: VideoMeter tool for YUV bitstreams. Technical Report acticom-02-001, Acticom - mobile networks, Germany (2002)
2. Video Quality Experts Group (VQEG), <http://www.its.bldrdoc.gov/vqeg/>
3. ITU-R: Methodology for the Subjective Assessment of the Quality of Television Pictures. ITU-R Recommendation BT.500-10 (2000)
4. Klaue, J., Rathke, B., Wolisz, A.: EvalVid - A Framework for Video Transmission and Quality Evaluation. In: Computer Performance Evaluation/TOOLS 2003, Illinois, USA, pp. 255–272 (2003)
5. VideoLAN – VLC media player, <http://www.videolan.org/>
6. Triple-play Services Quality of Experience (QoE) Requirements. Technical Report TR-126, DSL Forum (2006)
7. Venkataraman, M., Sengupta, S., Chatterjee, M., Neogi, R.: Towards a Video QoE Definition in Converged Networks. In: Second International Conference on Digital Telecommunications, p. 16 (2007)
8. Qiu, S., Rui, H., Zhang, L.: No-reference Perceptual Quality Assessment for Streaming Video Based on Simple End-to-end Network Measures. International conference on Networking and Services, p. 53 (2006)
9. Wattimena, F., Kooij, E., van Vugt, J., Ahmed, K.: Predicting the perceived quality of a first person shooter: the Quake IV G-model. In: Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support For Games, Singapore (2006)
10. ITU-T: Physical layer management for digital subscriber line (DSL) transceivers. ITU-T Recommendation G.997.1 (2006)
11. ITU-T: Asymmetric Digital Subscriber Line Transceiver 2 (ADSL2). ITU-T Recommendation G992.3 (2002)
12. Nedeve, N.: Analysis of the Impact of Impulse Noise in Digital Subscriber Line Systems. PhD thesis, University of Edinburgh (2003)
13. Tasaka, S., Watanabe, Y.: Real-Time Estimation of User-Level QoS in Audio-Video IP Transmission by Using Temporal and Spatial Quality. In: GLOBECOM, pp. 2661--2666 (2007)
14. Tao, S., Apostolopoulos, J., Guérin, R.: Real-time monitoring of video quality in IP networks. In: NOSSDAV, New York, pp. 129–134 (2005)
15. Moulin, F., Ouzzif, M., Zeddani, A., Gauthier, F.: Discrete-multitone-based ADSL and VDSL systems performance analysis in an impulse noise environment. IEE Science, Measurement and Technology 150(6) (2003)