A Video Frame Dropping Mechanism based on Audio Perception

Marco Furini
Computer Science Department
University of Piemonte Orientale
15100 Alessandria, Italy
Email: furini@mfn.unipmn.it

Vittorio Ghini
Computer Science Department
University of Bologna
40127 Bologna, Italy
Email: ghini@cs.unibo.it

Abstract— **Video streaming applications are more and more present in our life, but despite the advances of network technologies, several users experience QoS problems. This is mainly due to the high bandwidth requirements of these applications that contrasts with the network bandwidth limitation. To mitigate these QoS problems, video frame dropping mechanisms are often used for adapting the video stream to the network conditions. The selection of the video frames to drop is done considering the perceived quality of the video play out; audio perception is not considered in the selection process. In this paper we show that by taking into account only the video play out quality, audio problems arise very frequently. Hence, we propose a video frame dropping mechanism that takes into consideration the perceived quality of both audio and video play out. A comparison with other video frame dropping techniques is carried out and experimental results show that, although the video play out quality is similar, the audio play out quality is completely different. Our mechanism slightly affects the audio quality, while other techniques strongly affect it. Therefore, by using our mechanism, benefits are remarkable.**

I. Introduction

Networked multimedia applications are about to enter millions of private homes for entertainment and communication purposes and, thanks to the advances of network technologies and to the growing availability of digital contents, a large increase of such applications is expected in the near future.

Users will be able to access such multimedia applications from almost everywhere using portable devices: students can access the Net to get University lessons (or the last TV-show) using a simple notebook; commuters can watch the latest news using a palm device while being on a train; smart phones can be used to watch the preferred cartoons series while sitting on a bench in a public park. These are only some examples, but the combination of wireless technologies (Wi-Fi, Edge/GPRS, 3G), portable devices and bandwidth availability makes available multimedia applications from almost everywhere.

Unfortunately, in many cases the QoS achieved by these applications is not satisfactory. The main reason of these QoS problems is the bandwidth availability that, although less limited than in recent years, is still not sufficient for supporting several types of multimedia applications. For instance, if we consider the traffic produced by an audio-video streaming application (the most prominent multimedia application in the current Internet scenario), we can notice that it has high bandwidth requirements and significant bitrate variability: two characteristics that fight with the best-effort nature of the Internet. Further, the *last mile problem* should be not underestimated, as many users use low-bandwidth technologies to access the Net.

Hence, in networks where bandwidth is constrained or in best-effort networks, it may be not possible to deliver video streams to clients without incurring loss of data and hence the service should be denied. However, in some cases clients may choose to receive an imperfect quality of the video stream (a video with occasional frame losses), instead of having nothing. Needless to say, if the service is not for free, users should pay less for an imperfect QoS. Some kinds of video streams are willing to tolerate an imperfect QoS, as the overall quality is not compromised (university lessons, newsreport, TV-shows, to name a few), while some other videos are less tolerant (videomusic).

Among the techniques used to adapt the video stream transmission to the network conditions, the frame dropping is one of the most used [1], [2], [3], [4]. The reason is that these techniques are efficient and simple to use and, if well designed, they only slightly affect the quality of the delivered video. Several proposals have been done: *Lu* and *Christensen* [1] drop low priority video frames to enhance the overall quality of TCP-based video streaming applications; *Gurses et al.* [2] propose to drop video frames that are less important to human perception and hence, in MPEG videos, frames are discarded in order of importance (B-Frame, P-Frame and I-Frame); *Zhang et al.* [3] discard frames in order to minimize the likelihood of future frames being discarded; *Furini* and *Towsley* [4] use frame dropping techniques in a *diffserv* environments to propose a mechanism that provides the flexibility for the client to negotiate a tradeoff between bandwidth consumption and QoS with the server (and network).

The selection of the video frames to drop is usually done with the goal of maximizing the perceived quality of the video play out. While this is an important goal, results of extensive experiences have shown that audio is frequently perceived as the most important component of multimedia applications [5]. Hence, the perceived quality of the audio play out should be taken into consideration when discarding video frames,

0-7803-8798-8/04/$20.00 ©2004 IEEE

otherwise a good video play out quality might be coupled with a frustrating audio play out quality.

The contribution of this paper is the proposal of a video frame dropping mechanism that takes into consideration the perceived quality of both audio and video play out while selecting the video frames to drop. In essence, the frames selection process analyzes the audio information to find out all the silence periods in a video stream. These silence periods are then used to find all the associated video frames. Classic video frame dropping techniques are then applied to these video frames and hence the selection process identifies only video frames that are associated to silence. In this way, both audio and video play out are only slightly affected. A comparison with classic video frame dropping techniques is done and results show that the achieved video play out quality is very similar, while the audio play out quality is very different: if video frames are dropped without considering the perceived quality of the audio play out, the audio quality is strongly affected and may be frustrating. Conversely, our mechanism only slightly affects the audio play out quality. Hence, our approach provides remarkable QoS benefits.

The remainder of this paper is organized as follows. In Section II we present details and characteristics of our proposal, while in Section III we present a comparison between our approach and classic video frame dropping techniques. Conclusions and future work are presented in Section IV.

II. Selective Video Frame Discard Algorithm

In this section we present details of our proposal, a Selective Video Frame Discard (SVFD) mechanism which aims at selecting video frames to drop using both audio and video characteristics.

As we briefly mentioned, the frame dropping mechanisms proposed in literature may affect audio quality as the selection of video frames to drop is done focusing the attention only on the perceived quality of the video play out. As a result, there may be good perceived video play out quality, but the audio quality randomly depends on the selected video frames.

Our mechanism takes into consideration both audio and video play out quality. As depicted in Fig. 1, three steps are involved: i) a stream analysis is done to separate audio and video traces; ii) an audio analysis is performed to find out all the silence periods in the video stream and to determine the subset of video frames that are associated to these silence periods; iii) a video analysis is carried out to select the video frames to drop among those frames that are associated with silence. In the following we explain details of these steps.

A. *Stream Analysis*

A video stream is usually composed of two separate traces: one is related to the video part and the other regards the audio part. These two traces are then synchronized in order to have the classic audio/video effect. If we look at the composition of each trace, we can notice that a video trace is composed of a sequence of frames (video frames) and an audio trace is composed of a number of audio samples. The number of video frames and the number of audio samples depends on the video stream characteristics: for instance a NTSC video has usually 29.97 frames per second (fps), while a PAL video has usually 25 fps[1]; the number of audio samples depends on the used audio quality (44.100 samples per second provide good audio quality). Both the number of frames per second and the number of audio samples per second will be used in the audio/video analysis as described in the following.

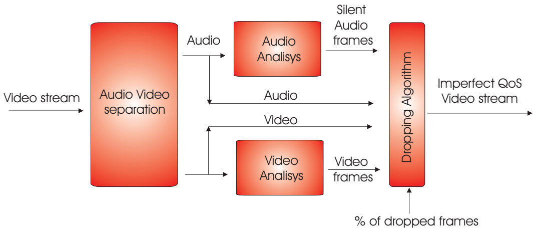

Fig. 1. Steps to obtain an imperfect QoS video stream.

B. *Audio Analysis*

The audio stream analysis is the fundamental part of our SVFD mechanism, as it detects all the silence periods present in a video stream. These silent periods will be later used to find the associated video frames.

To find a silence period in an audio signal, silence detector algorithm has to be used. In its simplest form the silence detection can be a magnitude based decision: the silence detector algorithm compares the magnitude of the signal against a preset threshold and if a percentage of the data is smaller than the threshold, silence is declared. Although the magnitude based algorithm has fairly mediocre performance in the presence of any background noise, it does not require much complexity. The Robust Audio Tool (RAT) uses a similar approach, where the threshold is automatically adjusted according to the audio characteristics [6]. Although more sophisticated approach may be used to find silence periods, we used the RAT approach and results were satisfactory.

Silence periods are an important component of a video stream as they are present massively. By using the number of frames per second and the number of audio samples per second, it is possible to identify the video frames that are associated with silence periods (from here on, we call these frames *silent* video frames). The subset composed of silent video frames will be later used by the video analysis in the selection process.

Table I shows the silence periods we found in some video streams we analyzed. We analyzed video streams with different characteristics: a cartoon (*The Simpsons*), a *newsreport*, a *talk-show* and a TV-movie (*24*).

Silence lengths are also very interesting to analyze. Fig. 2 shows the length of the silence periods we found in the analyzed streams and the frequency of these silence periods. For instance, the *24-series* has 40% of the silent periods

[1] NTSC and PAL are two television systems: the former is mainly used in US and Japan, while the latter is mainly used in Europe.

Video Trace	Total Length (sec)	Silent Period (sec - [%])
The Simpsons	1295	209 [16%]
24 (TV-Series)	2450	1241 [51%]
Newsreport	1914	651 [34%]
talk-show	693	182 [26%]

TABLE I
CHARACTERISTICS OF THE ANALYZED VIDEO TRACE.

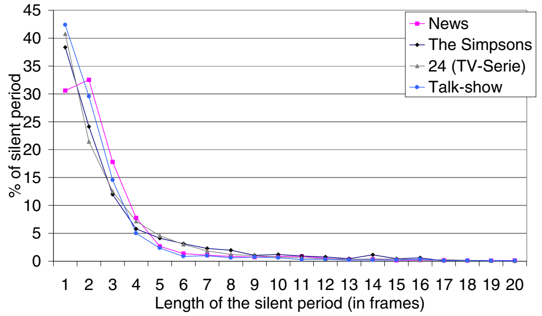

Fig. 2. Analysis of the silent periods.

associated with a single video frame (33.3 ms long), while 0.75% of the silent periods is associated with a sequence of ten consecutive video frames (333 ms long). The behavior is similar for all the analyzed traces and the percentage of silent periods decreases while increasing the length of the silent.

The reason of such a large number of short silence periods is explained in Figure 3, where a graphic representation of an audio signal is presented. In particular, Fig. 3 shows the energy of the sound obtained when a character of *The Simpsons* says 'MISTER HAMMOCK'. Note that there isn't a noticeable silence between the two words, but there is a silence period of 67 ms while saying the word 'MISTER' (between the pronunciation of the syllable 'MIS' and the syllable 'TER').

Fig. 4 shows the same audio trace with the silent period shortened from 67 to 34 ms and experimental evaluation confirmed that audio perception is not affected. The reason of removing exactly 33ms is due to the temporal length of a single video frame. This length is computed in the stream analysis (section II-A). Note that it is fundamental to shorten the audio in blocks, where every block corresponds to the temporal length of a video frame. In this way, both the audio and the video traces are shortened of the same time quantity and hence audio-video synchronization is not compromised. In fact, the goal of our mechanism is not the shortening of the audio trace (in this case more that 33 ms could have been removed), but is the dropping of video frames associated with silence. If a video frame falls in this 67ms silence, it can be removed without affecting to the audio perception. Hence, since each video frame lasts 33ms, the silence periods may be shortened in blocks of 33ms. For instance, Fig. 5 shows the video frames associated with the audio trace of Fig. 3. In this case one video frame is associated with the silent period. The video analysis will then decide whether this frame has to be dropped or not. Figure 6 shows a possible situation where the silent video frame has been dropped. Since the dropped frame falls in a silent period, the audio perception is not affected.

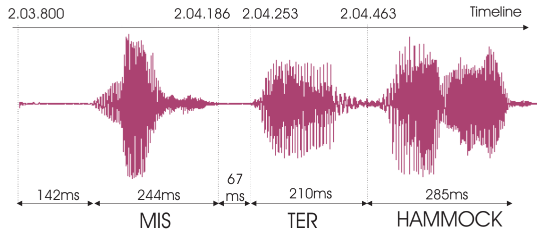

Fig. 3. The Simpsons. Audio signal while saying 'MISTER HAMMOCK'.

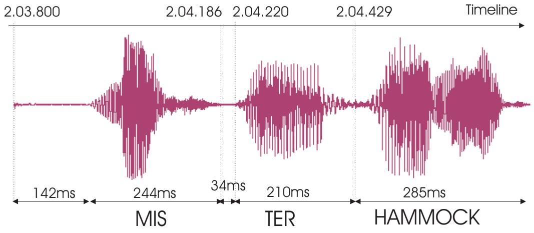

Fig. 4. The Simpsons. Audio signal while saying 'MISTER HAMMOCK'. Here the silence period is shortened of 33 ms.

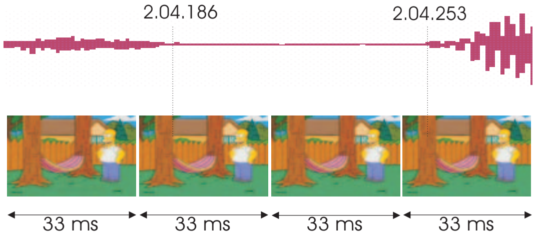

Fig. 5. Audio-Video association.

C. Video Analysis

In addition to the perceived audio quality, the video play out quality also plays an important role. Hence, the video frames selection has to take into account the QoS degradation. To better understand the dropping mechanism, we recall that a video stream is composed of several video frames and that each video frame may be independently decoded or may be decoded only with the help of other video frames. This

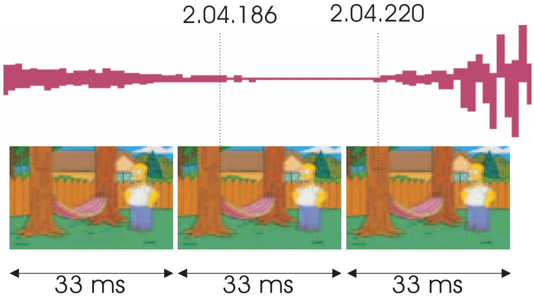

Fig. 6. A video frame associated with the silent period is dropped.

characteristic depends on the used encoding video technique: *intra*-frame or *inter*-frames. The former group (Motion JPEG is an example of this group) produces video frames that can be independently decoded, while the latter group (MPEG is an example of this group) produces video frames that cannot be independently decoded (hence, the discarding of some frames may result in the impossibility of decoding other frames).

Dropping video frame techniques [1], [2], [3], [4] take the encoding mechanism into consideration when deciding the video frames to drop and aim at minimizing the QoS degradation of the perceived video play out quality. For this reason, the frames selection process of our mechanism uses some of the dropping policies proposed in [3], [4]. Namely, for intra-frame encoded videos we use: **Discard Frame with distance** λ **(D(**λ**))**: The algorithm uses λ as a parameter that indicates the minimum distance between discarded frames. Unfortunately, there is no way to suggest the optimal value of the λ parameter, as it is affected by the characteristics of the considered video. Hence, different values of λ should be tested in order to select the best one. For inter-frames encoded videos our mechanism uses **Discard Third** P **Frame (DP3)** and **Discard** B **Frame (DB)**. DP3 discards only the P3-type of frame (and all the frames that depend on it), while DB discards only the B frames of the video.

It is important to point out that the above dropping algorithms are applied to silent video frames, while in [1], [2], [3], [4] dropping algorithms are applied to the entire set of video frames. By applying them to silent frames, the effects on the audio play out quality are mitigated.

The selection of the best dropping algorithm depends on the achieved video play out quality. However, it is worth noting that it is difficult to precisely define the perceived quality of the video play out; for this reason, cost functions are usually used to establish the perceived quality and therefore cost functions are used to compare different dropping algorithms. Roughly, a cost function analyzes the modified video stream and provides a cost value that represents the QoS degradation (a small cost value corresponds to little Qos degradation). An interesting cost function is proposed in [3]: it penalizes frame dropping mechanisms that drop neighboring frames as consecutive dropped frames may be more likely noticed by a user. This cost function takes two aspects into consideration: the length of a sequence of consecutive discarded frames and the distance between two adjacent but non-consecutive discarded frames. It assigns a cost c_j to a discarded frame j depending on whether it belongs to a sequence of consecutive discarded frames or not. If frame j belongs to a sequence of consecutive discarded frames, the cost is l_j, if the frame j is the l_j^{th} consecutively discarded frame in the sequence. Otherwise the cost is given by $1+1/\sqrt{d_j}$, where d_j represents the distance from the previous discarded frame. More details about this cost function can be found in [3].

In this paper we use this function to account for the perceived video play out quality and therefore we use it to compare different video frames dropping algorithms.

By discarding a percentage (requested either by the server, the client or the network) of the silent video frames, the final stream is ready to be delivered towards the client.

III. Experimental Results

To evaluate the benefits of our approach, we compare situations where the video frames selection process is done with or without considering audio information. It is to note that we focus on video streams that are likely to be watched in scenarios with bandwidth limitation (using a portable device while waiting for bus, or on a train while commuting). In such a scenario, users are not very focused on video play out quality and are willing to accept a lower video QoS if they can pay less for the service. Their goal is the entertainment, information or infotainment. For this reason, we use different types of video stream (cartoon, entertainment programs, TV-movie and NewsReport). It is also to point out that in such a scenario, audio quality is much more important as users usually use headset devices that lead the audio information to be very important.

To be more general as possible, we consider videos encoded with intra-frame technique (namely, Motion JPEG) and videos encoded with inter-frames technique (namely, MPEG). For each video stream, we produce several imperfect QoS video streams, dropping a percentage of video frames from 1% to 10%. Each imperfect QoS video stream is produced six times, as six different dropping policies are tested (half of them uses audio information in the video frames selection process). For each applied policy we compute: i) the cost of the dropping (using the cost function described in section II-C) and ii) the number of non-silent dropped video frames (not associated with silence).

This investigation allows us to compare the behavior of the different used policies. In fact, the cost value is used to compare the video play out quality, while the number of non-silent dropped frames is used to compare the audio play out quality (roughly, it can be seen as the number of audio problems that the user will experience).

A. Video Streams Properties

The analyzed video streams are encoded with 29.97 frames per second and have a resolution of either 352x240 or 320x240 pixels. The associated audio is two-channels with 44.100 samples per second in each channel.

In Table I we already showed the characteristics of the analyzed video streams (The Simpsons, 24, a talk-show and a newsreport). We recall that we refer to silent video frames as the video frames that are associated with silence, while we refer to the other video frames as non-silent video frames (or non-silent frames for short).

B. Intra-Frame Encoded Videos

In Motion JPEG videos, we use three different policies in order to select video frames to drop: RND, D(2) and D(5). The RND policy randomly selects video frames to drop; D(2) means that the minimum distance between two consecutive dropped video frames is two; D(5) is the same of D(2), but the

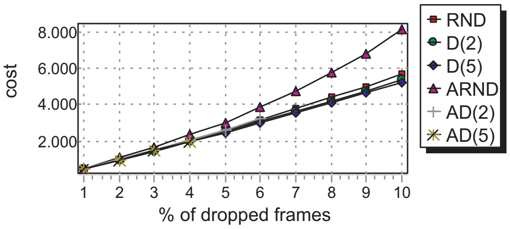

Fig. 7. Analysis of The Simpsons: cost of dropped frames.

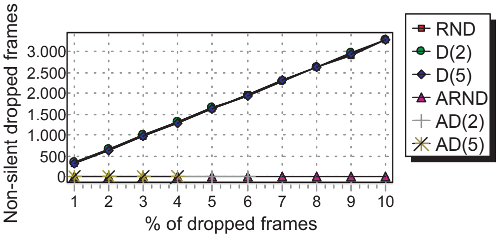

Fig. 8. Analysis of The Simpsons: Non-silent dropped video frames

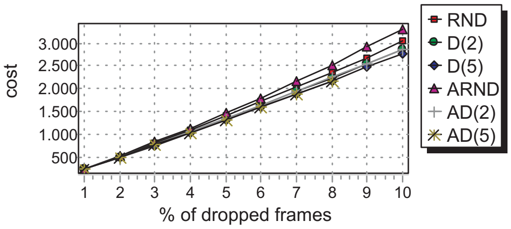

Fig. 9. Analysis of talk-show: cost of dropped frames.

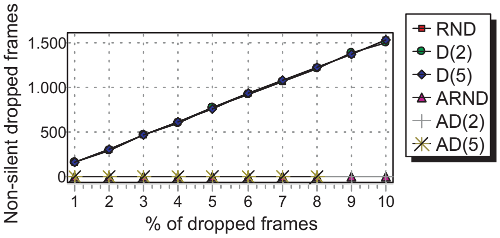

Fig. 10. Analysis of talk-show: Non-silent dropped video frames.

minimum distance is of five video frames. D(2) and D(5) avoid dropping sequence of consecutive video frames that would penalize the video play out quality. The above policies do not take into consideration audio information. If audio information are considered, the same three policies assume the name of ARND, AD(2) and AD(5).

In Figures 7-8 we present results obtained while analyzing an episode of *The Simpsons* cartoon series. While the cost values don't give us much information about the video play out quality, it is very useful to compare the different policies. In this case, all the policies perform similarly up to 5% and then the ARND provides a higher cost. The behavior of ARND is not surprising if we notice that this video stream has a small percentage of silence periods (16%). Hence, when it is necessary to drop considerable percentage of video frames, silence periods are shortened if not cancelled at all. This causes the dropping of consecutive video frames. It is also to note that AD(2) and AD(5) are not able to drop more than 7% of the video frames, as the number of silent video frames is not sufficient. However, it is to note that the video play out quality (which here corresponds to the cost value) is comparable for the AD(X) policies, regardless whether the audio information are used or not.

To complete the evaluation of our mechanism, we compute the number of non-silent dropped video frames. In Figure 8 we present the obtained results. Since ARND and AD(X) discard only silent video frames, their value is zero. Conversely, if audio information is not considered, the number of non-silent dropped video frames is considerable. For instance, if 5% of the video frames is dropped, the user perceives more than 1.500 audio problems and up to 3.000 if the percentage of dropped video frame is of 10%. Needless to say, the audio quality is strongly penalized.

By combining the above results, it is clear that our mechanism provides benefits as the video quality is similar and the audio quality is much better.

In Figures 9-10 we present results obtained while analyzing a talk-show. Also in this case, the cost values are similar for all the applied policies (also in this case ARND performs slightly worse than the others if the percentage is greater than 6%) and the audio problems are numerous if audio information were not used in the selection process.

C. Inter-Frames Encoded Videos

Due to the inter-frames dependencies, in MPEG videos we use policies that take into consideration the type of the dropped frames and hence, in addition to the selected dropped frame, also all the frames that depend on it are also dropped. Three different policies are used: RND, DP3 and DB. In this case the audio information are not used in selecting the video frames to drop. The RND policy randomly selects video frames to drop; DP3 drops only P3-Frames (and the ones that depends on it); DB discards only B-Frame. The same three policies are then used taking care of the audio information, and hence only silent video frames can be dropped; ARND, ADP3 and ADB are the names of these policies.

In Figure 11 we present results obtained from analyzing an episode of the series *24*. The cost is much higher than what experienced with Motion-JPEG videos, as the dependency mechanism causes the discard of consecutive video frames and the three policies perform very differently. By considering a single policy, we can notice that a similar cost is achieved regardless if the audio is considered or not: RND-ARND, DB-ADB and DP3-ADP3 perform similarly. As expected the two random policies perform worse than the others as the selected video frames may be of any types, causing the discard of long sequence of frames. (A)DP3 performs better than the random selection, but the best policies are DB-ADB. In fact,

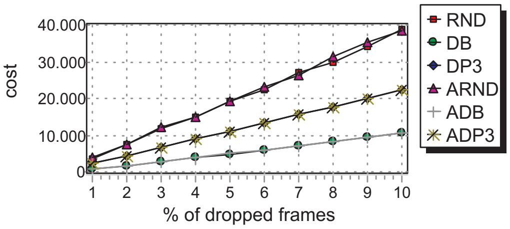

Fig. 11. Analysis of 24: cost of dropped frames.

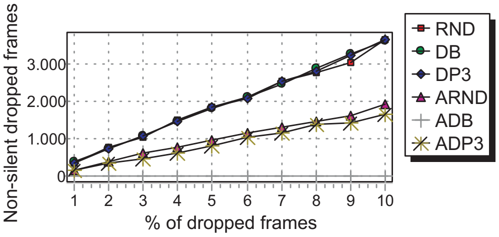

Fig. 12. Analysis of 24: Non-silent dropped video frames.

by discarding a B-Frame, the *domino*-effect does not happen and hence a long sequence of consecutive dropped video frames can never happen.

Figure 12 shows the number of non-silent dropped video frames. Note that, due to the *domino*-effect that may result when discarding a frame, it is possible that non-silent video frames are also discarded by our policies. In particular, ARND and ADP3 may discard non-silent video frames (the ADB policy drops only silent video frames). However, the non-silent discarded video frames are almost half of the ones discarded by policies that do not use audio information in selecting video frames to drop. Hence, also in this case, there are benefits in using our approach.

Figures 13-14 show similar results obtained from analyzing a Newsreport video.

D. Summary of Results

All the conducted experiments highlight that our mechanism does not affect the perceived video quality more than other techniques, but since our approach drops only video frames associated with silence, the overall user satisfaction is enhanced. In particular, we showed the impact of the dropped video frames on the audio quality. The effects on the audio quality is mitigated if our approach is used. Regarding the used policies, for Motion JPEG videos there is no much difference between the policies. Only the RND-ARND policies perform slightly worse than the others and hence they should be avoided. For MPEG video, the domino-effect heavily affects the results and hence the B-Frames policy should be used.

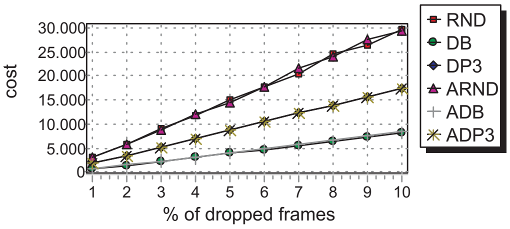

Fig. 13. Analysis of NewsReport: cost of dropped frames.

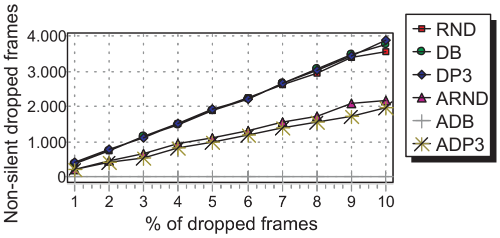

Fig. 14. Analysis of NewsReport: Non-silent dropped video frames.

IV. Conclusions and Future Work

In this paper we proposed an approach to select video frames to drop, using both the perceived audio and video play out quality in the selection process.

The evaluation of our approach has been done analyzing several and different types of video streams. A comparison is done with techniques that do not use audio information in the video frames selection process. Results showed that the perceived video play out quality is very similar regardless of the use of the audio information, but the perceived audio play out quality is much better if audio information are used in selecting video frames to drop. Hence, our approach provides remarkable benefits as it does not penalize the video quality and it mitigates the effects on the audio quality.

We are currently working on dropping algorithms for MPEG-4 encoded videos and we are analyzing benefits of our approach in diff-serv environments where bandwidth may be allocated in advance.

Acknowledgment

This work has been partially supported by the WEBMINDS FIRB project of the Italian MIUR.

References

[1] Y. Lu, K.J.Christensen, “Using Selective Discard to Improve Real-Time Video Quality on an Ethernet Local Area Network”, *International Journal of Network Management*, Vol. 9, 1999, pp.106-117.

[2] E. Gurses, G.B.Akar, N. Akar, “Selective Frame Discarding for Video Streaming in TCP/IP Networks”, in *Proceedings of the 13th IEEE Packet Video Workshop 2003*, April 2003, Nantes, France.

[3] Z.L. Zhang, S. Nelakuditi, R. Aggarwa, R. P. Tsang, “Efficient Server Selective Frame Discard Algorithms for Stored Video Delivery over Resource Constrained Networks”, *Journal of Real-Time Imaging*. 2000

[4] M.Furini, D. Towsley, “Real-Time traffic Transmission Over the Internet”, *IEEE Transaction on Multimedia*, 3(1), pp. 33-40, March 2001.

[5] V. Hardman, M.A. Sasse, I. Kouvelas, “Successful Multi-Party Audio Communication over the Internet”, in *Communications of the ACM* Vol. 41, 1998, pp.74-80.

[6] M. Roccetti, V. Ghini, G. Pau, P. Salomoni, M. Bonfigli, “Design and Experimental Evaluation of an Adaptive Playout Delay Control Mechanism for Packetized Audio for Use over the Internet”, *Multimedia Tools and Applications*, 14(1), 2001, pp.23-53.