

POLITECNICO DI TORINO  
Repository ISTITUZIONALE

A distributed video streaming platform supporting multiple description coding

*Original*

A distributed video streaming platform supporting multiple description coding / Baccaglioni, Enrico; Olmo, Gabriella; Grangetto, M.; Mirabelli, M.; Vinciarelli, E.. - (2010). (Intervento presentato al convegno ST Microelectronics STreaming Day 2010 tenutosi a Udine, Italy nel September).

*Availability:*

This version is available at: 11583/2377055 since:

*Publisher:*

*Published*

DOI:

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# A DISTRIBUTED VIDEO STREAMING PLATFORM SUPPORTING MULTIPLE DESCRIPTION CODING

Enrico Baccaglini\*, Gabriella Olmo\*, Marco Grangetto $\diamond$ , Mario Mirabelli $\ddagger$ , Enrico Vinciarelli $\ddagger$

\* Dip. di Elettronica  
Politecnico di Torino  
C.so Duca degli Abruzzi, 24  
10129 Torino, Italy

name.surname@polito.it

$\diamond$  Dip. di Informatica  
Università di Torino  
C.so Svizzera, 185  
10149 Torino, Italy

marco.grangetto@unito.it

$\ddagger$ Telecom Italia  
Research and Trends Dept.  
Via Reiss Romoli, 274  
10148 Torino, Italy

name.surname@telecomitalia.it

## ABSTRACT

This paper presents a video streaming system making use of H.264/AVC and multiple description coding. A complete architecture has been implemented, including the encoding, distribution and decoding processes, starting from H.264/AVC and using the RTSP and RTP protocols. These modules have been integrated in the *DynamicTV* prototype, developed by Telecom Italia Lab for distributed, multi server video streaming based on RTP/UDP. This platform has enabled realistic performance evaluation for both pre-encoded video distribution and live streaming, so allowing to assess the costs and benefits of a multiple description scheme for resilient video transmission.

## 1. INTRODUCTION

In modern multimedia scenarios, users may want to download or stream video data using terminals equipped with different capabilities in terms of power consumption, memory, computational resources and visual resolution. The contents may be accessed using broadband networks such as DSLs, optical cable or WiMax, but also full mobility GPRS/UMTS/HSDPA networks. Consequently, scalability is a key feature. The multimedia data quality should be matched to the visual and computational capabilities of the terminal at hand. Moreover, the signal received after transmission on unreliable networks should exhibit graceful degradation capabilities, so as to enable the decoder to achieve different quality levels, depending on the amount of correctly received information. Multiple description coding (MDC) is a possible tool to cope with this challenging context. In MDC, several non hierarchical, independently decodable representations of the same data (descriptions) are generated, yielding

mutually refinable information. Such descriptions are suitable for transmission over independent paths, and usually have the same importance as for the quality of the recovered data. Several methods for the generation of MD have been proposed, some of which are tailored for H.264/AVC video data. All methods achieve robustness at the expenses of some degradation in the compression efficiency; clearly the added redundancy should be carefully matched to the actual network conditions.

In this paper, we implement a full video streaming system making use of H.264/AVC and the MDC algorithm first proposed in [1]. This system, which encompasses the whole encoding, distribution and decoding chain, has been integrated in the *DynamicTV* prototype [2], developed by Telecom Italia Lab for distributed, multi server video streaming based on RTP/UDP. This platform has enabled realistic performance evaluation for both pre-encoded video distribution and live streaming, so allowing to assess the costs and benefits of a multiple description scheme for resilient video transmission.

The major contributions of the present work are the prototyping and performance evaluation of MDC in a full streaming application. This has allowed us to perform a realistic evaluation of MDC, taking into account the several impairments stemming from real-life transmission as opposed to the simulation results available in [1]. MDC streaming has been obtained by exploiting the compatibility of the selected technique with H.264/AVC, which permits to use the standard RTP/UDP protocol stack to transport the video packets. This yields datagram best-effort transmission, characterized by low complexity and low protocol overhead, but does not provide any guarantee of service. However, the intrinsic robustness of MDC makes it possible to rely on this protocol stack while maintaining satisfactory resilience properties. More-

---

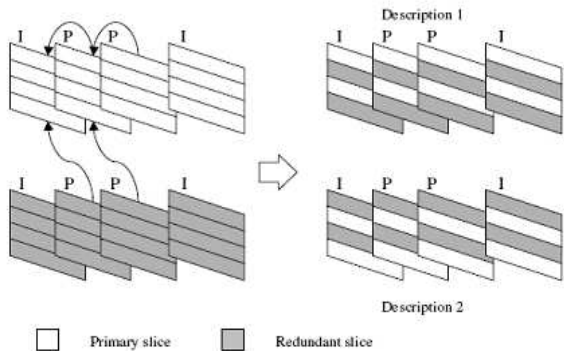
\*E.Baccaglini is now with STMicroelectronics, C. Olivetti 2, Agrate, Milan, Italy.

over, the H.264/AVC compatibility allows to use the standard RTSP protocol for media session signalling and codec initialization.

The rest of the paper is organized as follows. In Sec. 2 the MDC method used in this scheme is described. In Sec. 3 we briefly describe the *DynamicTV* platform. In Sec. 4, we detail the architecture of the implemented system. Finally, in Sec. 5 we show the experimental results and in Sec. 6 we draw the conclusions.

## 2. THE MDC ALGORITHM

In MDC, several non hierarchical, independently decodable representations of the same data (descriptions) are generated, yielding mutually refinable information. Such descriptions are suitable for transmission over independent paths, and usually have the same importance as for the quality of the recovered data. In this paper, we address the special case of two descriptions. The MDC algorithm described in [1] aims at creating two balanced descriptions, each of which is an H.264/AVC compliant bitstream. To this purpose, the concept of primary and redundant slices defined in the H.264/AVC standard [3] are used as shown in Fig. 1. The encoder generates the stream composed of



**Fig. 1.** Description generation with 4 slices per frame and 2 video descriptions.

the primary slices (white boxes in Fig. 1) at higher rate/quality along with the corresponding redundant slices (gray boxes) obtained by reducing the coding rate, e.g. by increasing the quantization parameter. According to the H.264/AVC standard the redundant slice are always predicted from the primary pictures; this choice limits the coding cost for redundant slices but on the other hand makes them dependent on the primary ones. The primary stream is encoded at high-rate  $R_H$  whereas the redundant stream is encoded at

the low-rate  $R_L$ . Primary and redundant slices are multiplexed in a single H.264/AVC bitstream. This latter bitstream can be separated into two balanced video descriptions so that each slice is coded at high-rate in one description and at low-rate in the other. When the primary representation of a slice is lost, but the redundant counterpart can be decoded, the decoder replaces the samples of the primary slice with the redundant representation.

At the decoder side, if both descriptions of a given slice are received, only the primary representation is retained and decoded (we refer to this as central quality decoding), whereas the redundant one is discarded. On the other hand, if a description is lost, the received one is a compliant H.264/AVC bitstream, containing a mixture of primary and redundant representations. This can still be decoded yielding inferior quality (side quality) due to the error drift injected by the decoding of some redundant slices. The redundant slice is obtained employing a larger quantization parameter than that of the primary representation. The value of the redundant QP allows one to shape the amount of coding redundancy inserted in the compressed bitstream. The rate/distortion optimization of this MDC tool has been addressed in [1]. The more redundancy  $\rho = R_L/(R_H + R_L)$  we insert in the encoding process, the more the two descriptions are similar. The parameter  $\rho$  can be set in the range  $[0;0.5]$ . If  $\rho = 0.5$ , the two description are very similar and the primary and redundant slice are coded with the same QP. No gain is achieved when both descriptions are received so that this value should be used only when the descriptions are transmitted over a network with high packet loss rate. On the other hand, a low redundancy should be set if the network is reliable.

## 3. THE DYNAMICTV PLATFORM

DynamicTV is a prototype platform developed in the research lab of Telecom Italia as an evolution of the current IPTV platform. The DynamicTV concept is based on the idea that the technical evolution may enhance the user TV watching experience by providing not only a more comprehensive choice (contents accessible via the Broadband channel are virtually infinite) but most important, a way to take the user from a pure passive TV consumption to a more interactive, on-demand, non-linear experience.

A potential for active TV experience is now available: users should be allowed to access contents that fit their attitudes from any provider, over any broadband network, at any time, on their television (Over-the-Top IPTV). Dynamic TV outlines the basic elements of the

new *Inter-tainment* paradigm, which builds on an active role of the user and on attitude-centered fruition. Novel 3D user interfaces, based on the real-world interaction or on exploring challenging metaphors are explored to allow alternative ways of consulting and discovering contents. The challenge is to design a new TV experience combining iTV features and 3D opportunities in a mixed 3D/media playing environment. Guaranteeing a good quality of service when retrieving any video content from anywhere and by means of heterogeneous access technologies and terminal devices represents a great research challenge. The system proposed in this paper aims at providing a robust streaming core for the IPTV platform, able to guarantee a good performance in presence of heterogeneous and best effort and potentially unreliable access technologies.

#### 4. DESCRIPTION OF THE IMPLEMENTED STREAMING ARCHITECTURE

##### 4.1. The MDC encoding library

The MDC encoding library has been developed as plug-in for the open-source VideoLAN VLC media player [4]. The encoding library has been obtained adding to the standard H.264/AVC encoder the possibility to create a redundant stream at a given rate  $R_L$ . This has been achieved by exploiting the H.264/AVC redundant picture coding option along with a modified rate controller. This latter represents a key feature of the propose prototype. In fact, previous performance evaluations of the adopted MDC technique [1] were based on fixed quantization parameter encoding. Clearly, when using MDC in a real streaming system, it is mandatory to adopt a rate control mechanism to shape the video coding rate on the fly. The developed MDC encoder includes two rate controllers that, given the total rate budget  $R_t = R_L + R_H$  and the desired value of  $\rho$ , predict different values of the quantization parameters to be used for the primary and the redundant slices, respectively.

The encoder library is distributed as pre-compiled binary file as shared object. The plug-in outputs two H.264/AVC compatible video streams made of primary and redundant slices of the input video source. From the VLC command line it is possible to specify many encoder parameters such as the output rate  $R_t$ , the MDC redundancy  $\rho$ , the Intra Period and the number of macroblocks per slice, etc. The video description can be stored using the mp4 file format or streamed live over the network using the RTP/UDP protocol. Given that  $R_t$  should be less or equal to the

input video rate, the encoder generates the descriptions using the constraint  $R_t = R_H + R_L = R_H + (\rho \cdot R_t)$ .

Finally, by exploiting the rich set of modules available in VLC the developed MDC coding library can be used to re-encode video from almost any format, including live encoding from live sources.

##### 4.2. MDC mixer

The real implementation of a MDC streaming system require a critical step on the decoder side, i.e. the synchronization of the two video descriptions. After synchronization occurs a simple preprocessing is used to forward the best representation of each slice to a standard H.264/AVC decoder. This task has been accomplished developing the MDC mixer, whose functional blocks are shown in Fig. 2. The MDC mixer is able to receive standard RTSP requests from the final player and behaves as a RTSP/RTP proxy towards and from the video servers providing the video description. Two separate RTSP sessions are setup on different video servers and the incoming RTP streams are synchronized so as to merge them into a single RTP stream toward a standard player. This latter decodes a standard H.264/AVC video and can be unaware of the underlying MDC encoding and streaming process.

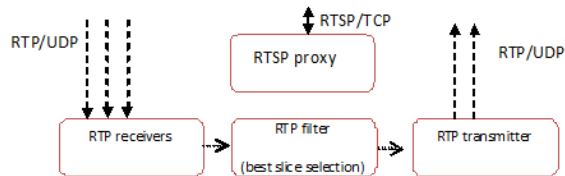


Fig. 2. Mixer block diagram.

##### 4.3. Darwin Streaming Server

Once the MDC streams have been coded and saved into mp4 file format by the encoder, they can be copied to video servers for video-on-demand services. To achieve this objective, we adopted Darwin Streaming Server (DSS) which is an open-source application released by Apple [5]. This tools supports RTSP/RTP streaming of pre-encoded mp4 files containing H.264/AVC video and AAC audio tracks. Here, we adopted DSS to store and stream pre-coded MDC video files. Alternatively, the developed encoder can behave as live RTP source, and can be interfaced to DSS by means of an SDP file, thus supporting a live streaming service.

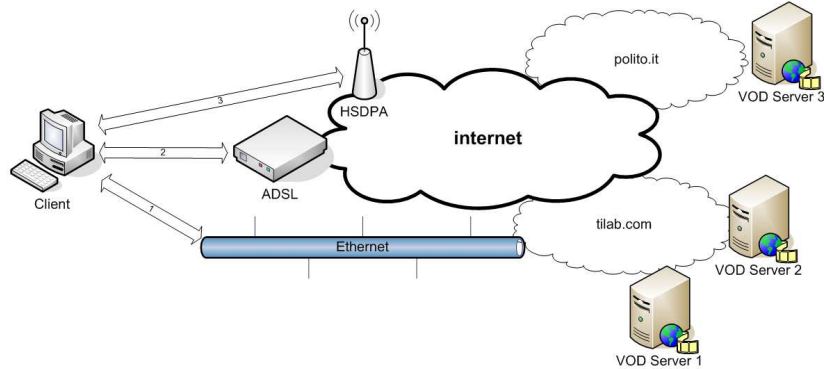


Fig. 3. Network topology

## 5. EXPERIMENTAL RESULTS

The goal of these measurements is to acquire general knowledge about the impact of MDC in a realistic environment by trying out a diverse set of existing technologies. The measurement scenarios were both on the local area networks (100 Mbps LAN) and across the public broadband network of Telecom Italia as shown in Fig.3. The client can receive the video from three different video servers. Two of them are on the local tilab.com Intranet and the other one is hosted by Politecnico. The client can access the public network over both a fixed broadband access (12 Mbps ADSL) and a mobile broadband access (7 Mbps HSDPA). In all the tests, the client receives the streams from the Politecnico video server and from one video server in the local Intranet. The Politecnico server is connected through 14 hops while the Tilab servers can be reached in 4 hops.

The measurement techniques used are:

- Network protocol layer tracing. In order to monitor the traffic of the MDC applications, a network layer snooping tool Wireshark [6] has been adopted to analyze the IP level traffic received at client side. Post-processing of the captured network data can provide information about loss rates at the client side.
- Application instrumentation. In commercial players, objective evaluation of the application performance is seldom possible. However, in our study, we were able to compare the received video with the original one. Once all frames have been saved by the mixer, a post-processing algorithm over the captured frames provides information about the video quality perceived by the final user by means of the PSNR metric.

We tested different video sequences downloaded from

*iTunes Movie Trailers* [7], with different resolution and duration. Here, we report the results for the *Made for each other* sequence,  $640 \times 360$  pixels, 24fps, which encompasses about 3300 frames. The original video sequence has a rate of approximately 1300 kbps and is coded using H.264/AVC Main Profile @ L3.0 with 2 reference frames. In order to allow fair comparison between SDC (traditional single description H.264/AVC coding) and MDC, the total output rate  $R_t$  is set to 1100 kbps. For SDC this is all devoted to video data, while for MDC each description has a rate of 550 kbps including both video data and redundancy. As for this latter parameter, we set  $\rho$  to 0.2 and 0.05.

The first column of Table 1 reports the PSNR values of MDC and SDC over LAN/ADSL/HSDPA. The transmission over LAN is lossless, representing an upper-bound on the achievable quality for our settings. In this case, the SDC transmission, which introduces no redundancy, allows to achieve the best performance. Insertion of unused redundancy in the video data is detrimental when no losses occur, but these results show that the performance degradation is limited (about 0.3 dB). The gap between MDC central quality and SDC depends on  $\rho$  and, in general, increases as the encoding rate decreases [1]. Hence, we expect that this gap would be larger with a lower  $R_t$ .

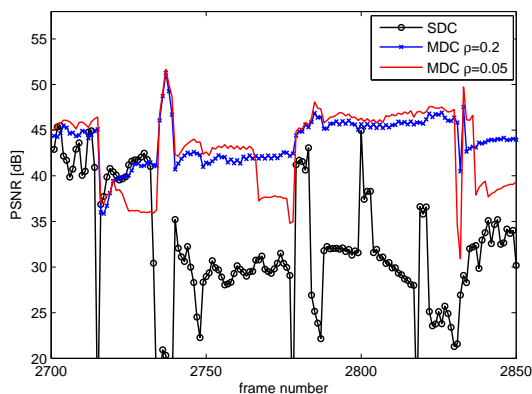
MDC proves its efficiency in lossy scenarios. In fact, as for ADSL transmission, MDC achieves a gain up to 6 dB when the packet loss probability is about 3%. It is worth noticing that the two video descriptions are routed over two different paths, and encompass different packet loss rates. When the application is required to operate across the public internet, there is a significant differences between the number of hops separating the client from server in Tilab (which is small) and the client from server in Polito (which is high). Thus the connection between client and server

**Table 1.** Expected PSNR, packet loss rate and sequence error rate.

network		$E\{\text{PSNR}\}$ [dB]	p(loss) [%]	seq.err. [%]
LAN	SDC	44.86	0	0
	MDC $\rho = 0.2$	44.50	0,0	0,0
	MDC $\rho = 0.05$	44.60	0,0	0,0
ADSL	SDC	36.85	3.3	6
	MDC $\rho = 0.2$	42.64	2,7.2	12.8,14.2
	MDC $\rho = 0.05$	40.61	5.4,2.8	5.6,2.8
HSDPA	SDC	30.28	7.8	6
	MDC $\rho = 0.2$	41.83	17.7,0.3	12.4,13.3
	MDC $\rho = 0.05$	36.78	15.5,0.3	12.5,4

in Polito has a higher probability of packet loss rates.

The gain over traditional SDC is significant and even more sharp for the HSDPA scenario where the mean packet loss rate is about 8%. The MDC scheme allows the receiver to perceive a mean quality of approximately 40 dB. Comparing the redundancy of MDC in the ADSL and HSDPA we highlight that a higher  $\rho$  allows to better recover from packet losses. In ADSL, the gain between the two redundancy levels is about 2 dB and increases to 5 dB for HSDPA.



**Fig. 4.** PSNR vs. frame number for the tested sequence.

In Fig.4 we report the PSNR of each video frame in the range [2700-2850] for the ADSL scenario in which the packet loss rate is approximately 4%. MDC can rely on two video streams to recover the frames so that, in case of loss of one of the two representations, the other may be available and this allows the decoder to recover the video (e.g. around frame 2700). Furthermore, MDC can reduce the video quality fluctuations, which lead to a very annoying display visualization. The impairment between different values of  $\rho$  can be

seen around frame 2800 where the two MDC transmissions undergo the same losses and the performance with  $\rho = 0.05$  are slightly higher than with  $\rho = 0.2$ .

## 6. CONCLUSIONS

In this paper we tested a MDC scheme based on the redundant slice feature of H.264/AVC over different network scenarios provided by Telecom Italia Lab. Results show that MDC can significantly outperform traditional coding when packet losses occur, with a gain of up to 6 dB when the network is particularly unreliable (HSDPA). In case of lossless transmission, the MDC redundancy reduces the mean quality by approximately only 0.3 dB. Future work will include analysis of live content streaming by means of SDP protocol.

## 7. REFERENCES

- [1] T. Tillo, M. Grangetto, and G. Olmo, "Redundant slice optimal allocation for H.264 multiple description coding," *IEEE Trans. on Circ. and Syst. for Video Tech.*, vol. 18, no. 1, pp. 59 – 70, Jan 2008.
- [2] R. Simeoni, M. Geymonat, E. Guercio, M. Perrero, A. Rapp, F. Tesauri, and R. Montanari, "Where have you ended up today? Dynamic TV and the entertainment paradigm," in *Proc. of EuroITV*, July 2008.
- [3] T. Wiegand, G.J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circ. and Syst. for Video Tech.*, vol. 13, no. 7, pp. 560–576, Jul 2003.
- [4] "Video Lan Client Player," <http://www.videolan.org>.
- [5] "Apple Darwin Streaming Server," <http://developer.apple.com/opensource/server/streaming/>.
- [6] "Wireshark," <http://www.wireshark.org/>.
- [7] "Apple iTunes Trailer," <http://www.apple.com/trailers/>.