



HAL
open science

On the limit of fountain MDC codes for video Peer-To-Peer networks

Guillaume Smith, Pierre-Ugo Tournoux, Roksana Boreli, Jérôme Lacan,
Emmanuel Lochin

► To cite this version:

Guillaume Smith, Pierre-Ugo Tournoux, Roksana Boreli, Jérôme Lacan, Emmanuel Lochin. On the limit of fountain MDC codes for video Peer-To-Peer networks. 2012 IEEE Thirteenth International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Jun 2012, San Francisco, United States. pp.1-6, 10.1109/WoWMoM.2012.6263799 . hal-02553140

HAL Id: hal-02553140

<https://hal.science/hal-02553140>

Submitted on 24 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 5899

To cite this document: Smith, Guillaume and Tournoux, Pierre-Ugo and Boreli, Roksana and Lacan, Jérôme and Lochin, Emmanuel *On the Limit of Fountain MDC Codes for Video Peer-To-Peer Networks*. (2012) In: 1st IEEE WoWMoM Workshop on Video Everywhere, 25-28 Jun 2012, San Francisco, United States.

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@inp-toulouse.fr

On the Limit of Fountain MDC Codes for Video Peer-To-Peer Networks

Guillaume Smith
ISAE - Université de Toulouse, France
NICTA - UNSW, Australia

Pierre-Ugo Tournoux
and Roksana Boreli
NICTA, Australia

Jérôme Lacan
and Emmanuel Lochin
ISAE - Université de Toulouse, France

Abstract—Video streaming for heterogeneous types of devices, where nodes have different devices characteristics in terms of computational capacity and display, is usually handled by encoding the video with different qualities. This is not well suited for Peer-To-Peer (P2P) systems, as a single peer group can only share content of the same quality, thus limiting the peer group size and efficiency. To address this problem, several existing works propose the use of Multiple Descriptions Coding (MDC). The concept of this type of video codec is to split a video in a number of descriptions which can be used on their own, or aggregated to improve the global quality of the video. Unfortunately existing MDC codes are not flexible, as the video is split in a defined number of descriptions. In this paper, we focus on the practical feasibility of using a Fountain MDC code with properties similar to existing Fountain erasure codes, including the ability to create any number of descriptions when needed (on the fly). We perform simulations using selected pictures to assess the feasibility of using these codes, knowing that they should improve the availability of the video pieces in a P2P system and hence the video streaming quality. We observe that, although this idea seems promising, the evaluated benefits, demonstrated by the PSNR values, are limited when used in a real P2P video streaming system.

I. INTRODUCTION

Video content has become hugely popular on Internet [1] and the resulting video traffic adds new constraints to the network in terms of capacity while some video applications, *e.g.* streaming, have strong delay constraints. Furthermore, the network is composed of heterogeneous types of devices ranging from a smartphone with low computational capacity and small screen, to a powerful computer with a HD screen, therefore having different video quality requirements while sharing the same network.

The large number of end-user devices accessing video content can also overload the video server capacity. To address this problem, a proposed solution is to take benefit of a Peer-To-Peer (P2P) network to decrease the load on the server by sharing the data (referred to as content pieces) between all the nodes as in [2]. P2P networks are known to provide high throughput and ability to cope with failure, churn and heterogeneous node's capacity. In the particular context of video streaming, P2P solutions must also provide sequentiality to ensure that chunks which are due for playout are not incomplete or missing. Providing sequentiality limits re-buffering, or frames skip. In file sharing P2P such as BitTorrent, the high throughput and robustness are mainly due to the diversity of the chunks available on peers which is a consequence of

the rarest-first chunk selection algorithm. The sequentiality prevents the use of rarest-first and reduce the diversity. As it was proven in [3], there is no system satisfying those three constraints and users must be able to cope with varying throughput and incomplete chunks.

To respond to the different quality of the receivers, a solution is to use Multiple Description Coding (MDC) codes [4] which split a video into n descriptions. Each description brings out information about the video, thus the more descriptions a user receives, the better the quality of the video is. A receiver is able to download the full quality video if the n descriptions are available, but high churn rate, congestion in the underlying network or link layer losses may prevent the receiver to complete the download of chunks in time. If a P2P streaming solution integrates an MDC code, partial chunks translate to less than n descriptions allowing a graceful quality degradation (*i.e.* lower quality of video without stopping or skipping frames [4]). If the use of MDC code allows a trade-off between video quality and buffering time, it remains that the low chunk diversity impacts on the throughput and robustness.

If MDC codes are able to produce an infinite amount of descriptions while allowing to reconstruct high quality video with n of those, seeders could increase the chunk diversity by generating different description for each peer. Towards this goal, this paper studies the feasibility of such a Fountain MDC codes. We propose a practical scheme and assess the quality of reconstructed video compared to standard MDC. Finally we discuss the limitations of such a scheme, its implication in terms of complexity and its integration in a video codec.

Our contributions include the novel concept of combined MDC and Fountain codes for P2P video streaming, a proposal for a specific code and the evaluation of the performance of this code on a selected set of pictures.

The paper is organised in the following way. In Section 2, we provide a background on P2P video streaming, Fountain codes and MDC codes. In Section 3, Fountain MDC code are presented and then simulated in Section 4. Finally, we discuss and conclude this paper in Section 5.

II. BACKGROUND ON P2P STREAMING

During the past few years, the adaptation of P2P networks to the context of streaming applications have received a significant interest from the research community. This section

review the key design concept and the use of erasure coding and the adapted video codec.

A. Network structure and chunk selection

In live P2P streaming, randomly connected mesh networks have been promoted [5] as they allow path diversity, churn resilience and a simple construction and maintenance of the topology. As in file-sharing P2P, the random mesh structure implies that chunks diversity impacts directly on the capacity of peers to help each other and achieve high throughput. The downside is that contrary to tree-structured networks, the limited availability of future content in live streaming applications as well as the need for sequentiality to achieve smooth playback affects the chunks diversity. This issue has been partially addressed by the use of playout buffers and by pushing missing chunks randomly with a probability proportional to the playback time of the chunk [6].

B. Rateless erasure code

Usage of source or peer-based rateless erasure codes¹ have been proposed to increase chunk diversity without increasing the delay [7], [8]. For instance, the authors of [9] propose a source-based encoding with Fountain codes. One or more sources creates and pushes encoded chunks built from k source chunks. These encoded chunks have the same size than the source ones. The characteristic of rateless codes is that they can generate an infinite amount of distinct encoded chunks [10]. A given block can be decoded when slightly more than k coded chunks are received. In this case, the diversity is increased as source chunks have multiple representations in the network and a peer can accept coded chunks describing the same block pushed by multiple source.

Random linear coding is used to perform peer-based coding (*i.e.* network coding) such as [8] where peers re-encode the various coded chunks available. This allows to increase a bit further the diversity and reduce the overhead and control messages. The downside of erasure codes is that chunks cannot be played until the full block has been decoded².

C. MDC codes

As previously described, Multiple Description Coding is made to split a single video stream in multiple streams, called descriptions. Different kind of MDC codes exist: Scalable Video Coding (SVC) requires the first i descriptions to decode the $i + 1^{th}$ description and increase the quality of the video [11]. MDC codes that produce independent descriptions are furthermore interesting³ as they tolerate the loss of any description [4]. This is an ideal solution for multiple-tree based P2P networks with the transmission of one description per tree. In random mesh networks, they cope well with churn [12] and the heterogeneity of peers connectivity [13].

¹The encoding part can use different types of operations as the XOR, linear combinations or operations on Galois fields.

²Obviously, encoding should be media aware *e.g.* if a block is as a function of the GOP (Group Of Pictures) size, it is sufficient to decode a block to ensure that the GOP will be played without errors (as in [9]).

³Until now, SVC codecs provide a better ratio between fidelity and compression than independent MDC codes.

D. Combinations of techniques

Both rateless erasure codes and MDC video codecs improve the overall Quality of Experience (QoE) of a video streaming on a P2P network and many proposals use both schemes conjointly [9], [14], [15]. In these works, the video is encoded in multiple descriptions using an MDC codec. Then, on each description a rateless code is used. However in these papers, the techniques are used one after the other and not combined. In other words, a user first needs to decode the rateless code and then, use the MDC code to display the video.

The remaining of this paper investigates the design of a *Fountain MDC code* which inherits from the properties of both techniques, *i.e.*: to be able to create an infinite number of encoded packets (Fountain property) which are all useful on their own and improve each other (MDC property).

III. OUR PROPOSAL: THE FOUNTAIN MDC CODE

The Fountain MDC code would enable the creation of an infinite number of descriptions of a video, creating these descriptions when needed, *i.e.* on-the-fly. From the receiver point of view, each received description would increase the quality of the video (as a classical MDC code would) and when a given number of them is received, the resulting video would have a sufficient quality (as a Fountain code, a file can be decoded when the receiver receives a little more information than the size of the file). In a P2P network, this code would allow a seeder to send new descriptions to each node which is in contact with this seeder. This new description is then useful to any other nodes in the network, which thus may increase the chunk diversity on the P2P network. Furthermore, a peer which does not have a sufficient amount of descriptions to decode the full quality may be able to make linear combination of the ones received creating another description (similarly to Network Coding [16]), also improving the chunk diversity. Finally as a standard MDC code, it would allow heterogeneous devices to play the same content with different quality. This would allow to adapt the content as a function of the screen resolution of the device or the network capacity, by downloading more or less descriptions.

To create one of this Fountain MDC codes, we focus our effort on creating a Fountain code which would have the MDC property to increase the video quality for each new received description. Thus, we have specifically worked on pictures, which would represent the different I frames created by a video codec as with *h.264* coding scheme [17].

Usually, Fountain codes are introduced at the network layer, so in this context, it would be after the codec at the source side. However in this case, a receiver would have to decode the Fountain code to use the information. As our purpose is to be able to use information not totally decoded, we create the Fountain code at the application layer to be able to use this encoded information, and more precisely before the compression which is not a linear process.

The whole process from the recorded picture to the displayed one at the receiver side follows four steps as illustrated in Figure 1. At the source side, the first step is to split the

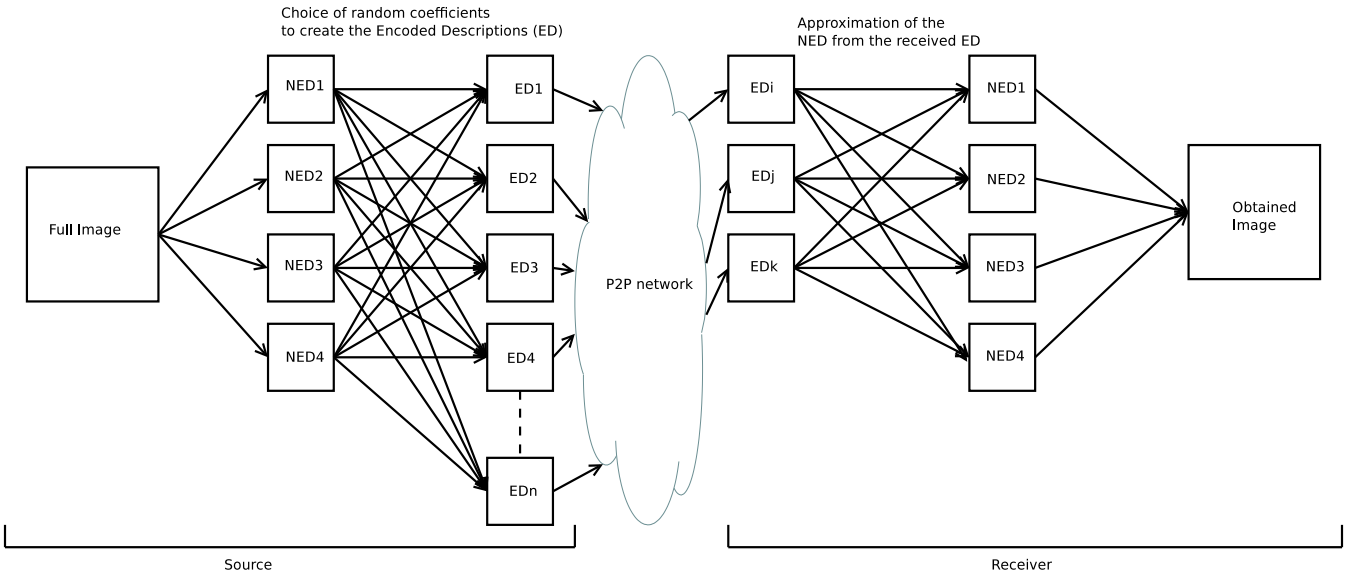


Fig. 1: The whole process when for example three ED are received.

picture in a number of non-encoded descriptions (NEDs), then the creation of the encoded descriptions (EDs) which can be made on-the-fly by making linear combinations of the previous NEDs. The different streams are then sent to the users. At the receiver side, first a certain amount of streams are received. From the received EDs, the next step is to approximate the NEDs from the linear combination obtained to finally rebuild the whole picture. We precisely describe each step of the process in the following.

A. Creation of four non encoded descriptions

From the picture, four descriptions are created from a spatial subdivision by taking one pixel on four per description from all the $2 * 2$ blocks of pixels as in Figure 2. These four descriptions will be referred to as non-encoded descriptions (NEDs). As explained in [18], this spatial subdivision allows to create four descriptions with a low computational power and without creating a whole new video codec.

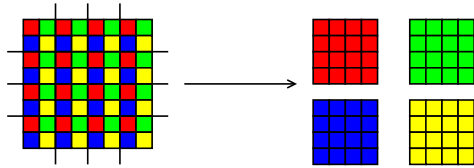


Fig. 2: Creating the four non-encoded descriptions from the picture

B. Creation of the infinite number of descriptions

1) *In theory*: The four NEDs are used to create the infinite number of encoded descriptions (EDs) by performing the Fountain process as in Figure 1. The constraint at this stage is to have a useful description on its own after the encoding.

For this purpose, we are doing random linear combinations of the four NEDs with coefficients which sum is equal to 1:

$$ED_i = \alpha_i * NED_1 + \beta_i * NED_2 + \gamma_i * NED_3 + \phi_i * NED_4$$

with $\alpha_i + \beta_i + \gamma_i + \phi_i = 1$. With this process, all the pixels after the encoding still have a value between 0 and 255 and are a barycentre of the pixels from the NEDs. The choices of the coefficients will be explained in Section III-B2.

When the EDs are created, the different pixels do not have integer values, but are float numbers (due to the coefficients which are not integers). To be sent on a network or to be used by a standard codec (as *JPEG* for pictures or *h.264* for video), the different pixels values are rounded to the closest integer. This is the first stage where errors appear in the process as the numbers are approximated.

2) *Choice of the coefficients*: We choose two types of random coefficients to create the EDs from the NEDs. The first choice is to create the four random coefficients close to each other. This choice allows to have a good description of the full image when one ED is received, but because of the rounding to obtain integer, the different EDs created with this process can be close to each other. As the difference between them can be small, it is more complicated to obtain new information when a new ED is received. In practice, we choose to take randomly the coefficients between 1 and 4 and then normalise them so that the sum is one. In the following, this configuration is denoted *small coefficients* configuration.

Then we choose a second type of random coefficients: one coefficient is dominant and the others are small in comparison. This choice describes well one NED in an ED, while little information about the other NEDs is present. We denote in this case that an ED_i describes an NED_j if the coefficient used for the NED_j is the dominant one. In practice, the dominant coefficient is around 20 times higher than the small

ones. Then, they are also normalized so that the sum totals one. For the case a peer receives four EDs describing the four original NEDs, we obtained nearly the same results as receiving directly the four NEDs. Thus in the following, we only study the worst case for the receiver: each received ED describes the same NED_j. This corresponds to the worst case as every time, the same NED is well characterized while the receiver does not have precise information about the others. In the remainder of the paper, this configuration is denoted *large coefficient* configuration. In a P2P system, to increase the availability of chunks, a user should not be forced to differentiate between two descriptions built from the same frame. In the *large coefficient* case, this worst case scenario is quite likely, *i.e.* it is likely that two or more EDs will describe the same NED, as the receiver does not choose a description based on the way to encode it, but only on the frame it was built from.

Finally, we also introduce a threshold condition on the ED when it is created. When the coefficients are chosen, the average error due to the rounding is computed for each pixel and each color. If this value of the error is higher than:

- $\frac{3}{\sum \text{coefficients}}$ for the *small coefficients* configuration and
- $\frac{10}{\sum \text{coefficients}}$ for the *large coefficients*;

the ED created is replaced with a newly created one. Applying this threshold condition reduces the amount of errors introduced by the rounding when EDs are created.

C. Decoding the streams

1) *Overview*: If a codec is used, the first step at the receiver side is to decompress the stream in a picture where all pixels are described by three bytes. Then, depending on the number of received EDs, the process to decode them can differ but after the decoding process, four NEDs are approximated to finally reconstruct the whole picture.

2) *Operations done to decode the Fountain code*: In this section, we explain the operations done to approximate the full image as a function of the number of ED received:

- 1) If only one ED is received, we just use the value obtained for the four pixels;
- 2) If two EDs are received, the operation used for the approximation is the barycentre. Using the same notations as in Section III-B1, we approximate the pixels in each NED as follows:

$$NED_1 = \frac{\alpha_1 * ED_1 + \alpha_2 * ED_2}{\alpha_1 + \alpha_2};$$

$$NED_2 = \frac{\beta_1 * ED_1 + \beta_2 * ED_2}{\beta_1 + \beta_2};$$

$$NED_3 = \frac{\gamma_1 * ED_1 + \gamma_2 * ED_2}{\gamma_1 + \gamma_2};$$

$$NED_4 = \frac{\phi_1 * ED_1 + \phi_2 * ED_2}{\phi_1 + \phi_2};$$

- 3) If three EDs are received, we are also doing the barycentre method as previously but with three coefficients

instead of two. This solution is compared with a more complex one: knowing that the solutions are integers, we solve the Diophantine system with three equations and four unknowns by choosing the last value which gives the lowest difference with the received ED. Then, we use these solutions to determine directions. We do not use them directly as the rounding process creates to much inaccuracy to have a good solution. Finally, we add or subtract (depending on the direction obtained) the value $\frac{1}{\max(\alpha_1, \alpha_2, \alpha_3)}$ to the barycentre of NED₁ for example. We add or subtract this fraction value as the pixel can be equal to both these values and still have the same rounding in the linear operation. This fraction corresponds to the imprecision of the rounding for this specific NED;

- 4) To finish, if four EDs are received, we are doing the barycentre process and as an alternative, we are also solving the linear system made by four equations with four unknowns by inverting the matrix built from the random coefficients.

IV. RESULTS

Simulations are done with the well-known Lena picture which is a 512*512 pixels size in color (RGB) (see Figure 3). In Table I, the receiver receives the NEDs from the *JPEG* codec with a quality of 100% which still creates errors due to compression, as we can observe, if we compare them with the one in Table II in which the picture is not compressed. The results in both tables are close, the main difference is when the receiver gets four NEDs, the PSNR (Peak Signal-to-Noise Ratio) grows from 50dB to ∞. Note that for a video application, a PSNR equals to 50dB is already an excellent quality. These two tables are presented as references.

In Table III, we compute the average PSNR on 50 simulations, when the receiver gets one to four EDs which are not compressed by a codec, in order to analyse only the effect of the errors due to the rounding in the encoded process. The EDs are built with small or large coefficients, and with or without the threshold condition on the coefficients as explained in Section III-B2. As explained in the part III-C2, depending on the number of EDs received, different decoding algorithms are used.

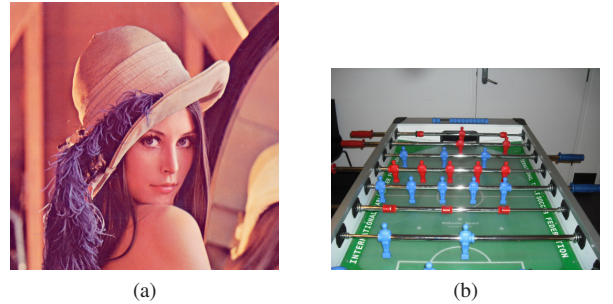


Fig. 3: Pictures used for the simulations

First, we can see in Table III that the *large coefficients*

configurations (with or without the threshold condition on the coefficients) do not bring any improvements when new EDs are received. The configurations with *large coefficients* are worst cases, *i.e.* when all the EDs are describing the same NED. This result shows that the *large coefficients* is equivalent to receive directly the NED. Actually, receiving two or more EDs describing the same NED do not improve the PSNR, which implies that in this configuration, the receiver cannot download any ED built from the frame. It has to choose one which brings new information, exactly as without the Fountain process.

Then, with the *small coefficients* configuration, we can first observe that using the threshold condition on the coefficients improves the average quality of the picture. Then the MDC property is verified: the more EDs are received, the better is the quality. But the improvement is quite low: from $31dB$ with one ED to $37.5dB$ with four EDs in the best case, furthermore the improvement is only equal to $0.5dB$ when three EDs are received compared to one.

Finally, when compared to the standard MDC (when the NEDs are directly received), the best configuration of the Fountain process brings out an improvement when only one ED is received (difference of $3dB$). However, receiving a new NED improves more the PSNR than receiving a new ED. Thus with three NEDs received, the PSNR obtained is equal to $35.5dB$ when the one obtained with the ED is equal to $31.5dB$. Finally if the four NEDs are received, the PSNR is infinite while with the EDs, it is only equal to $37.5dB$ on average.

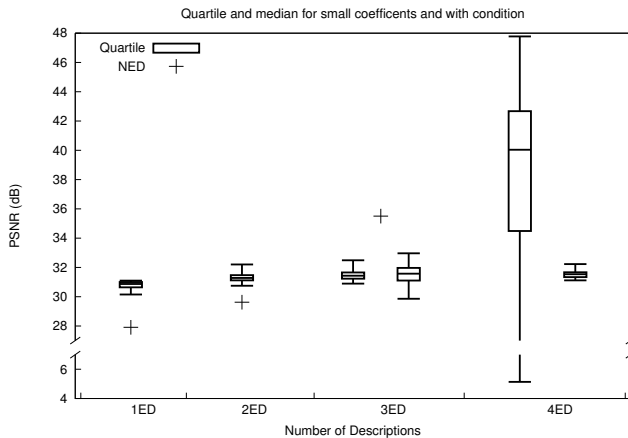


Fig. 4: Quartiles obtained when the small coefficients with the threshold condition configuration is used. When three EDs are received, the box on the left is obtained with the barycentre techniques, the one on the right with the Diophantine. When four EDs are received, the box on the left represent the matrix inversion, the one the right the barycentre. When four EDs are received, the value for the NEDs is infinite.

On average, we observed that the *small coefficients* configuration with threshold condition brings out the best results com-

1 NED	2 NEDs	3 NEDs	4 NEDs
27.88	29.56	35.37	50.30

TABLE I: PSNR (dB) for the NEDs with Codec

1 NED	2 NEDs	3 NEDs	4 NEDs
27.91	29.62	35.50	∞

TABLE II: PSNR (dB) for the NEDs without Codec

pared to the other case. Thus, to have a better understanding of the results obtained, Figure 4 shows the quartiles obtained with this configuration as a function of the method used and the number of received descriptions. The PSNR obtained for the NEDs are also plotted as a reference. Therefore we can see that the barycentre method has a very low variance, and in 100% of the case, it brings out better results when one or two EDs are received compared to NEDs receptions. But in all case it has a worse PSNR when three or four descriptions are received. Then when four EDs are obtained, to inverse the matrix shows a better PSNR than the barycentre method in more than 75% of cases, but in few cases, the decoded picture can be unusable (the minimum PSNR is equal to $5.1dB$).

As a final test, we use another picture of a table soccer which is composed by 3264×2448 pixels in color (see Figure 3). We made only five simulations with one configuration which is the *small coefficients* without threshold condition and without using a codec. The algorithms used to recompose the full picture at the receiver side is the barycentre one when one to three EDs are received, or the matrix inversion when four EDs are received. The results presented in the Table IV are the average PSNR, the worst and the best case obtained on the different simulations. The results are similar to the one obtained from Lena's picture, which tends to prove that the simulated results are not linked to the picture used.

Number of EDs	Average without threshold condition	Average with threshold condition
1 ED large coeff	29.36 (0.011)	29.10 (0.008)
1 ED small coeff	30.92 (0.007)	30.80 (0.008)
2 EDs large coeff	29.47 (0.008)	29.18 (0.005)
2 EDs small coeff	31.26 (0.009)	31.30 (0.009)
3 EDs large coeff	29.45 (0.006)	29.16 (0.005)
3 EDs Diophantine chooses the direction large coeff	27.34 (0.095)	25.21 (0.155)
3 EDs small coeff	31.39 (0.007)	31.47 (0.010)
3 EDs Diophantine chooses the direction small coeff	31.76 (0.014)	31.54 (0.022)
4 EDs inversion large coeff	26.84 (0.268)	26.46 (0.239)
4 EDs large coeff	29.49 (0.005)	29.16 (0.004)
4 EDs inversion small coeff	34.97 (0.224)	37.52 (0.227)
4 ED small coeff	31.42 (0.006)	31.53 (0.007)

TABLE III: PSNR (dB) for Lena picture without codec, summary, barycentre except if noticed (the coefficients of variation are in parenthesis)

Number of EDs	Average	Worst case	Best case
1 ED	33.45	33.23	33.56
2 EDs	33.79	33.48	34.25
3 EDs	33.85	33.55	34.21
4 EDs (inversion)	35.73	26.31	43.83

TABLE IV: PSNR (dB) for Table Soccer picture without codec, the configuration tested is the small coefficients without threshold condition.

V. CONCLUSION, DISCUSSION AND FUTURE WORK

Streaming a video over a P2P network composed of heterogeneous devices with different calculation, bandwidth and display characteristics, is a complex problem that has been tackled in several studies. MDC codes and rateless codes are two possible solutions and are sometimes used conjointly in some studies. However, MDC codes are not flexible and when combined with a rateless code, this code still has to be decoded to be useful. The Fountain MDC code introduced in this article combined both the property of the MDC code to improve quality for each received descriptions, and the rateless property in order to create an infinite number of descriptions on-the-fly. This kind of code would allow to increase the chunk diversity over a P2P network, i.e., its global throughput and robustness. In this paper, we assess the practical feasibility of the Fountain MDC code by proposing a low-computation one to estimate the gain in PSNR as a function of the number of received descriptions when used on pictures. Although the MDC and Fountain properties are achieved and the idea seems promising for the chunk diversity on a P2P network, we observe that the gain in PSNR obtained per new description is limited (denoising filtering recommended by MPEG-4 [19] is not used in this study to observe the capacity of the Fountain MDC code without artefact), which tends to limit its deployment.

Then, as our main goal is to study the feasibility of the Fountain MDC code, we did not investigate details of a potential video implementation. For future work, we still have to understand how to create the motion vectors for this type of codec. Actually, random ED are created, but they are finally closed to the original NED. Thus, we have to study the effect of the linear combination used to create the ED on the motion vector and more precisely, to assess whether we need to send the motion vector created from the NED or from the ED. Otherwise, as we are rebuilding the whole pictures, it could be relevant to use directly the motion vectors created from the whole video before the splitting process.

Finally concerning the improvement on a P2P network, we think that the Fountain MDC code could increase the availability of the different chunks on the network. We expect to drive measurements to verify this improvement. Furthermore, with this type of codes, a node in the network could also utilize network coding on the different descriptions it receives, knowing that network coding can usually improve the global performance of a P2P network as in [20]. A peer in the P2P network (which also watches the video) can first obtained different ED, and then make a random normalized

linear combination of them which becomes a new MDC encoded description. However it may imply new rounding, and consequently additional errors which have to be corrected.

REFERENCES

- [1] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. –, Aug. 2010. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2043164.1851194>
- [2] E. Setton, P. Baccichet, and B. Girod, "Peer-to-peer live multicast: A video perspective," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 25–38, Jan. 2008.
- [3] B. Fan, D. G. Andersen, M. Kaminsky, and K. Papagiannaki, "Balancing throughput, robustness, and in-order delivery in P2P VoD," in *Proc. CoNEXT*, Dec. 2010.
- [4] V. K. Goyal, "Multiple description coding: Compression meets the network," 2001.
- [5] N. Magharei and R. Rejaie, "Understanding mesh-based peer-to-peer streaming," pp. 10:1–10:6, 2006. [Online]. Available: <http://doi.acm.org/10.1145/1378191.1378204>
- [6] Y. Borghol, S. Ardon, A. Mahanti, and N. Carlsson, "Toward efficient on-demand streaming with bittorrent," in *IFIP Networking 2010*, L. N. i. C. S. M. Crovella et al., Ed. Springer, May 2010, pp. 53–66.
- [7] D. Jurca, J. Chakareski, J.-P. Wagner, and P. Frossard, "Enabling Adaptive Video Streaming in P2P Systems," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 108–114, 2007.
- [8] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, 2007.
- [9] J.-P. Wagner, J. Chakareski, and P. Frossard, "Streaming of scalable video from multiple servers using rateless codes," in *Multimedia and Expo, 2006 IEEE International Conference on*, July 2006, pp. 1501–1504.
- [10] D. MacKay, "Fountain codes," *Communications, IEE Proceedings-*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.
- [11] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.
- [12] S. Milani, S. Busato, and G. Calvagno, "Multiple description peer-to-peer video streaming using coalitional games," in *EURASIP, ser. European Signal Processing Conference (EUSIPCO 2011)*, 2011.
- [13] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, *P2P Video Live Streaming with MDC: Providing Incentives for Redistribution*. Ieee, 2007, pp. 48–51. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4284583>
- [14] P. Garbacki, D. H. J. Epema, J. Pouwelse, and M. van Steen, "Offloading servers with collaborative video on demand," in *Proceedings of the 7th international conference on Peer-to-peer systems*, ser. IPTPS'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 6–6. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1855641.1855647>
- [15] H. R. Oh and H. Song, "Mesh-pull-based p2p video streaming system using fountain codes," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, 31 2011-Aug. 4 2011, pp. 1–6.
- [16] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *Information Theory, IEEE Transactions on*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
- [17] I. Richardson, *H.264 and MPEG-4 video compression: video coding for next-generation multimedia*. Wiley, 2003. [Online]. Available: http://books.google.com.au/books?id=ECVV_G_qsxUC
- [18] A. Vitali, "Multiple description coding—a new technology for video streaming over the internet," *EBU Technical Review*, 2007.
- [19] G. Raja and M. J. Mirza, "In-loop deblocking filter for jvt h.264/avc," in *Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 235–240. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1365904.1365944>
- [20] M. Wang and B. Li, "Network coding in live peer-to-peer streaming," *Multimedia, IEEE Transactions on*, vol. 9, no. 8, pp. 1554–1567, Dec. 2007.