

Document downloaded from:

<http://hdl.handle.net/10251/58088>

This paper must be cited as:

Belda Ortega, R.; De Fez Lava, I.; Fraile Gil, F.; Arce Vila, P.; Guerri Cebollada, JC. (2014). Hybrid FLUTE/DASH video delivery over mobile wireless networks. Transactions on Emerging Telecommunications Technologies. 25(11):1070-1082. doi:10.1002/ett.2804.



The final publication is available at

<http://dx.doi.org/10.1002/ett.2804>

Copyright Wiley

Additional Information

Hybrid FLUTE/DASH Video Delivery over Mobile Wireless Networks

Román Belda, Ismael de Fez, Francisco Fraile, Pau Arce and Juan Carlos Guerri

Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València, Valencia, Spain

ABSTRACT

This paper describes how FLUTE (File Delivery over Unidirectional Transport) and DASH (Dynamic Adaptive Streaming over HTTP) can be used to provide mobile video streaming services over broadcast wireless networks. FLUTE is a multicast protocol for multimedia file download. In this proposal, the protocol is adapted to provide video streaming services in crowded environments. Thus, video is delivered over a single connection to all viewers, reducing the traffic in the network. FLUTE incorporates an AL-FEC (Application Layered – Forward Error Correction) mechanism in order to improve the reliability of the broadcast communication channel. For streaming applications, AL-FEC improves the relationship between the PSNR (Peak Signal-to-Noise Ratio) of the received video and the bandwidth allocated to the broadcast connection. The AL-FEC hereby presented applies simple unequal error protection schemes to favor the download of key frames. Furthermore, the proposal is based on the same video segmentation mechanism as DASH and therefore, clients can connect to a DASH repository to repair errors in the segments. This paper shows that FLUTE and DASH can be seamlessly integrated into a hybrid broadcast/unicast streaming technology, providing flexibility to trade off PSNR and bandwidth depending on the conditions of the mobile network.

KEYWORDS

File delivery over unidirectional transport (FLUTE), dynamic adaptive streaming over HTTP (DASH), media presentation description (MPD), video streaming, mobile wireless networks, unequal error protection (UEP).

1. INTRODUCTION

In recent years, IP networks have emerged as a delivery channel for mobile video services, fostering the appearance of different IP based video delivery technologies, each with different characteristics. These technologies can be classified according to the underlying IP method used to send the video. This way, unicast technologies send the video to each viewer on a separate IP connection. Oppositely, broadcast technologies establish a single connection from the source to all hosts connected to the network, regardless if the video is being watched or not at any particular host. Moreover, multicast technologies use a single connection, as in broadcast, but send the video only to interested viewers.

On the other hand, current Internet video services use technologies based on the Hypertext Transfer Protocol (HTTP), e.g. HTTP progressive download, pseudo-streaming or the new standard DASH. Due to the ever-

growing penetration of these services, this kind of traffic is becoming more and more prominent in IP networks.

However, unicast HTTP traffic can put a lot of pressure into the network. Thus, wireless network infrastructure may be overwhelmed by the amount of traffic generated by HTTP video services, especially when there are many simultaneous viewers. In these situations, IP video delivery technologies could benefit from a broadcast content delivery mechanism. With this in mind, this paper proposes the use of FLUTE to provide video services over wireless networks. Although neither HTTP nor FLUTE was originally meant for streaming video both technologies can provide support for video streaming applications if need be.

Our proposal is based on the use of a hybrid FLUTE/DASH network for video delivery services based on broadcast or multicast IP. This approach results rather innovative comparing it to current related work. Thus, it is worth highlighting the standards MBMS (Multimedia Broadcast /Multicast Service) [1]

and eMBMS (Evolved MBMS) [2] from the 3GPP Project, which propose the use of RTP (Real Time Transport Protocol) for video streaming and FLUTE to download files. Also, FLUTE is proposed to send information related to DASH (such as signaling or DASH segments). Unlike our proposal, this related work do not propose FLUTE to send video (which is quite novel) nor DASH as a video repair service. A related reference is [3], which presents an overview of the challenges of mobile video streaming, such as DASH over eMBMS. In fact, [3] explains the usage of FLUTE for transmitting DASH segments. In contrast to [3], this paper evaluates the system proposed and analyzes the use of different FEC code rates to protect the different types of video frames.

In this sense, the use of the same segmentation scheme in both protocols can yield innovative ways to distribute video segments in wireless networks with broadcast or multicast support.

The rest of the paper is structured as follows. The next section presents an overview of video coding. Section 3 explains the main technologies used in this work: FLUTE and DASH. Also, Section 4 describes the system proposed. Section 5 presents the methodology used to evaluate the proposal, which is analyzed in Section 6. Finally, the last section includes some final conclusions about the study as well as the future work.

2. VIDEO CODING OVERVIEW

Modern video encoders based on MPEG-4 family of video coding standards, such as H.264/AVC (Advanced Video Coding), exploit adjacent frames and nearby pixel correlation to reduce temporal (inter-frame) and spatial (intra-frame) redundancy as well as perceptually unimportant information. Thus, video frames are classified into I, P and B-frames, depending on the coding dependency. Following this scheme, video stream is grouped in a fully decodable Group of Pictures (GoP). Within a GoP, I-frames (intra-coded pictures) are encoded independently of any other frame in that GoP. Alternatively, P-frames (predictive coded pictures) use motion/estimation compensation based on information related to a previous frame. Finally, B-frames (bi-directionally predictive coded pictures) can reference previous and subsequent frames (backward and forward reference). Remark that this is a basic coding scheme, whereas state-of-the-art encoders support more advanced encoding combinations and techniques [4] [5]. According to this basic coding scheme, if an error is produced during transmission in any I-frame, this error

is propagated throughout the GoP due to inter-frame dependencies. However, errors in P or B-frames only affect dependent frames, causing lesser video distortions, as depicted in Fig. 1. Therefore, I-frames become more important than P or B-frames.

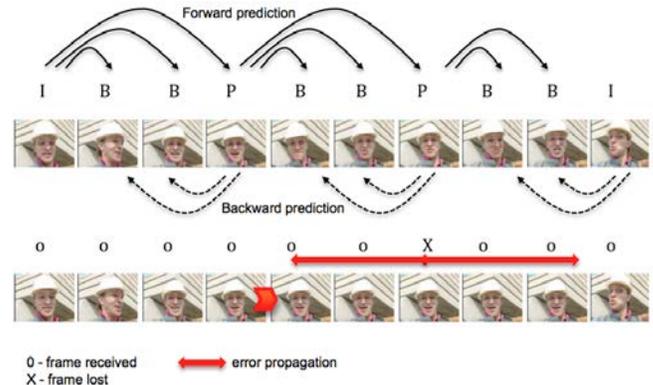


Fig. 1. Simplified video coding scheme.

Due to these relationships between frames within a GoP, especially for video streaming applications, the effect caused by wireless channel losses deteriorates the Quality of Experience (QoE) perceived by the viewer if these packet losses occur within I-frames. Some solutions proposed to minimize packet losses in wireless channels are based on Automatic Retransmission Request (ARQ) mechanisms. However, in real-time streaming services it is recommendable the use of other techniques to avoid stale packets. In this sense, Forward Error Correction (FEC) allows the receiver to recover some packets lost by means of sending parity packets in addition to video packets. FEC mechanisms sacrifice effective bandwidth in benefit of reducing errors and improving packet protection without the need of retransmissions. It is worth noting that not every frame must have the same level of protection. Thus, unequal error protection (UEP) has been proposed in several works regarding video streaming [6]. UEP can be achieved in video streaming at packet level (e.g. using data partitioning in H.264), frame level (I, P or B) and layer level in the case of multilayer scalable video coding. Taking into account the requirements of mobile devices and the limitations regarding available H.264 profiles and compatibility, the most suitable UEP mechanism turns out to be at frame level.

Regarding the user experience, users normally care more about seeing a standard definition video which plays smoothly than seeing a high definition video with slow starts, stuttering and re-buffering. This fact suggests that it is worth implementing those mechanisms for video streaming that reduce the start-up

latency and guarantee and improve, as much as possible, the quality and the smoothness perceived by the viewer.

3. TECHNOLOGIES

3.1. FLUTE

FLUTE, defined in RFC 6726 [7], is a protocol for the delivery of files over the IP multicast network service. The main characteristic of the protocol is that it provides reliability in the transmission, needed in multicast networks that do not guarantee the successful delivery of the packets. The three main mechanisms used by FLUTE to provide reliability are: AL-FEC (Application Layered – Forward Error Correction) to add redundancy and correct errors; retransmissions by means of data carousels to receive packets previously lost; and file repair sessions to request certain packets that have not been received.

FLUTE works over ALC (Asynchronous Layered Coding) [8], which is a protocol suitable for massively scalable, unidirectional and multicast content distribution services. Also, ALC is built over three building blocks: LCT (Layered Coding Transport), a congestion block and a FEC block. The transmission in FLUTE is based on delivery sessions, which are uniquely identified by the destination IP address and by an identifier called TSI (Transport Session Identifier). Also, each session contains one or several ALC/LCT channels, in which files are delivered. Each channel sends multimedia content in a certain port number with a certain transmission rate. The files sent are identified by a content location and by an object identifier called TOI (Transport Object Identifier). Moreover, the main characteristics of the files sent within a session are described in XML (Extensible Markup Language) through an in-band mechanism used by FLUTE called File Delivery Table (FDT). The FDT is sent together with the files through FLUTE packets called FDT Instances, with TOI equal to 0.

Regarding the transmission scheme, there are two main types of FLUTE sessions: file delivery sessions and file delivery carousels. In carousels, files are sent cyclically on a seamlessly endless loop. On the other hand, in FLUTE transmissions, each file (or object) is split into source blocks. Also, each source block is split into encoding symbols. To that extent, the FLUTE RFC [7] proposes an algorithm that splits the file into blocks and symbols in an efficient way. The payload of a FLUTE packet contains, at least, one encoding symbol. There are two kinds of encoding symbols: source and

parity symbols. The first ones conform the original data of the file, whereas the parity symbols are created from a combination of source symbols (and other parity symbols) through FEC encoding, in order to provide error correction techniques.

It is worth noting that in FLUTE the encoding process is carried out in each block and therefore, different blocks can use different AL-FEC codes. This allows providing different protection to different blocks, therefore it is possible to use UEP techniques through FLUTE in a simple way. This characteristic is one of the main points of the proposal presented in this paper.

3.2. DASH

HTTP streaming [9] and specifically DASH [10] is a new ISO standard for the transmission of on-demand and live content with time-shifting capabilities. DASH is based on multimedia file segmentation. Each multimedia file is encoded in different qualities and every quality file is split into small portions called segments. In order to access the whole multimedia content, clients select which quality of each segment they want to download.

DASH defines a manifest file, called *Media Presentation Description* (MPD), which describes the multimedia content, the different qualities and how the content is split into segments. A video service, for instance a live feed or a video on demand, is represented by a *media presentation*, which is a collection of time dependent media items, as shown in Fig. 2.

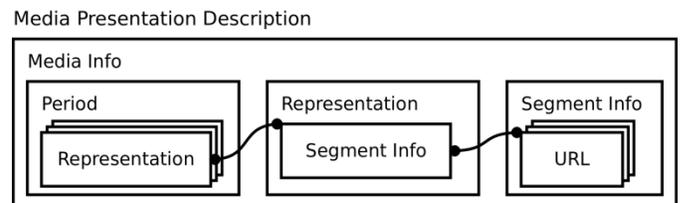


Fig. 2. DASH Media Presentation Description.

Media presentations are composed by a sequence of *periods*. Periods are time intervals along the duration of the media, which cannot overlap. Each period has different encoded alternatives, referred to as *representations*. For example, the same video of a certain period can be encoded at different bitrates, which, in turn, determine the quality of the playback. Thus, clients can select from multiple quality options. The selection mechanism depends on the target application. For instance, for live streaming it is convenient that clients switch to the representation that best fits the available bandwidth to avoid video

stuttering. Representations are sequences of at least one segment, which in turn has the access information to the segment itself.

Clients select the time when they want to download segment files, request the desired representation and switch between the representations. Since representations need to be independent of each other to allow stream switching, segments are composed by an integer number of GoPs.

4. SYSTEM OVERVIEW

The system proposed in this paper considers videos encoded with two different qualities, as depicted in Fig. 3. Each FLUTE channel streams a certain video with a specific quality. Moreover, each channel has an associated DASH server. Each DASH server contains a video encoded with the same quality as the video of the associated FLUTE channel. Once a fragment of the video is received through a FLUTE channel, clients can request frames lost to the DASH server.

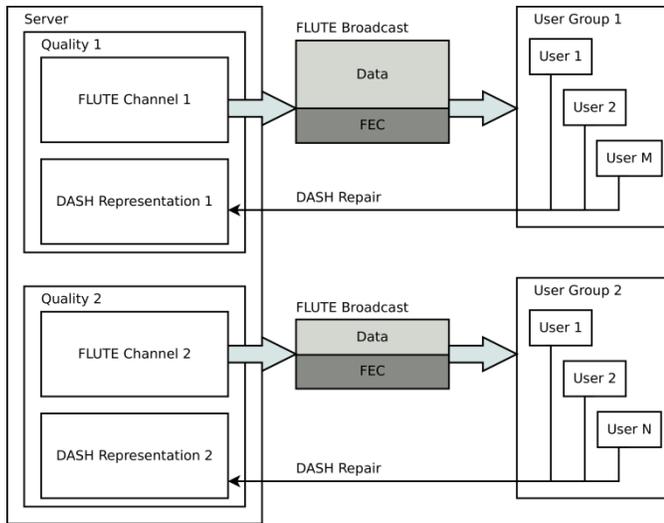


Fig. 3. Hybrid FLUTE/DASH scenario.

Clients connect to a specific FLUTE channel according to the packet error rate they detect. Thus, clients with a good connection (high bandwidth and low packet losses) connect to the channel that sends the video with the maximum quality. Those clients with high losses connect to the channel with less quality. This way, videos with high quality need less protection since they are consumed by clients with low losses, and vice versa. Thus, the FEC protection applied to the videos with less quality will be higher than the FEC protection applied to the videos with high quality: the higher the quality of a video, the less error-protected. The idea of

using different FLUTE channels with different FEC protection has been used in other solutions, such as [11], proposed by the authors of this paper.

Among the different kinds of video sources, this proposal takes into consideration live video streams. Moreover, it is assumed that clients know the time when the video transmission is carried out, so they are connected to the channel before the transmission starts. Other related solutions in the literature to highlight are [12], which analyzes on-demand video services using FLUTE; and [13], which proposes an approach for the streaming of videos using IP multicast (but not using FLUTE). Unlike these solutions, our proposal uses the FLUTE protocol in the provision of video streaming services.

The segmentation process is depicted in Fig. 4. After the sampling and the encoding processes the video is segmented according to a given segmentation policy, generating DASH segments and FLUTE blocks. The information related to the DASH segmentation is indicated in the MPD. Then, clients can access the segments through a FLUTE session and a DASH server. The flexibility regarding the segmentation of a video in FLUTE and DASH allows combining both technologies to download video frames.

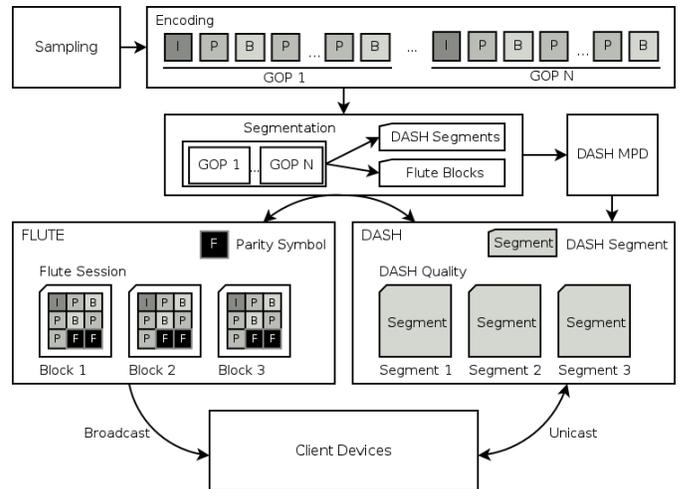


Fig. 4. Hybrid FLUTE/DASH video delivery architecture.

Each FLUTE block represents a video frame, and is formed by n encoding symbols: k source symbols and $n - k$ parity symbols. The number of parity symbols is determined by the FEC code rate, defined as k/n , so the less the code rate the higher the number of parity symbols. In our proposal, the AL-FEC block applies different AL-FEC rates to source symbols belonging to I, P or B-frames, that is, unequal error protection [6]. I-frames have more (or equal) protection (and therefore

more parity symbols) than P and B-frames, since I-frames are more important, as discussed above. Also, blocks of different frames can have different number of symbols, depending on the size of the frame. The code rate and the coding parameters regarding each block are included in a FLUTE extension header, in a format that depends on the type of coding used.

Clients connect to the FLUTE multicast session (before the server starts the transmission) and filter the packets belonging to the objects they want to download. After being connected for a moment clients know their packet error rate and can make decisions about the quality that better fits their network reliability.

Users receive the different symbols that compose each block and rebuild the corresponding GoPs. The parity symbols received are used to recover the source symbols lost. If after applying AL-FEC decoding some packets have not been recovered, clients are able to request only the frames lost to the DASH server through a unicast connection. As mentioned, clients know the information referred to the DASH server by means of the manifest file (MPD), which identifies the alternative locations for each segment in their respective segment information descriptions.

The way DASH carries out the segmentation is very flexible, and each segment could contain from only one video frame to several GoPs. In our proposal, we assume that there is a DASH segment per GoP [14], and each GoP is further divided into sub-segments. In any case, in the event of losses, a client can request to the DASH server either the entire GoP, or only the lost frames.

Clients start playing the video when a certain number of GoPs are received, thus providing low initial start-up latency. This number depends on the buffer of each client. In order to provide a continuous video playback, the transmission bandwidth has to be higher than or equal to the playback rate of the video. When clients are displaying the content of GoP n , they are receiving packets of following GoPs through the hybrid FLUTE/DASH network. It should be mentioned that, in the evaluation of the system proposed, we do not take into account the buffer needed to playback the video without stuttering, which is part of the future work. On the other hand, the FLUTE transmission rate and the number of requests to the DASH server depend on the available bandwidth to fulfill with the playback rate condition.

5. EVALUATION METHODOLOGY

In order to show the trade-off between bandwidth and video PSNR, we have evaluated the proposal using two sample videos created by the Blender Foundation [15]. The selected videos are: “Sintel”, an animation short; and “Tears of Steel”, a real/animated action short. We have extracted samples of 60 s from the middle of each video to avoid beginning and ending credits. Both videos have been encoded with two different resolutions and bitrates. Specifically, the resolutions considered have been 720p (HD) and 1080p (Full HD). Their frame rate is 24 fps and the GoP size is also 24, resulting GoP durations of 1 s. Table I shows the main characteristics of the videos evaluated.

TABLE I
CHARACTERISTICS OF THE VIDEOS TO EVALUATE

ID	Name	Resolution	Bitrate
Video 1	Sintel	720p	1500 kbps
Video 2	Sintel	1080p	4000 kbps
Video 3	Tears of Steel	720p	1500 kbps
Video 4	Tears of Steel	1080p	4000 kbps

Fig. 5 and Fig. 6 show the instantaneous bitrate of the videos used in this paper. Specifically, Fig. 5 shows the bitrate profile of Video 1 and Video 2 (“Sintel”), and Fig. 6 shows Video 3 and Video 4 (“Tears of Steel”). As depicted in the graphs, both encoding bitrate profiles are quite similar, the only difference is the encoding bitrate. In the figures we can see how the bitrate changes over the time, with peak values of bitrate until 8000 kb. Although the average bitrate is the same for both videos, the bitrate fluctuation for “Sintel” is higher.

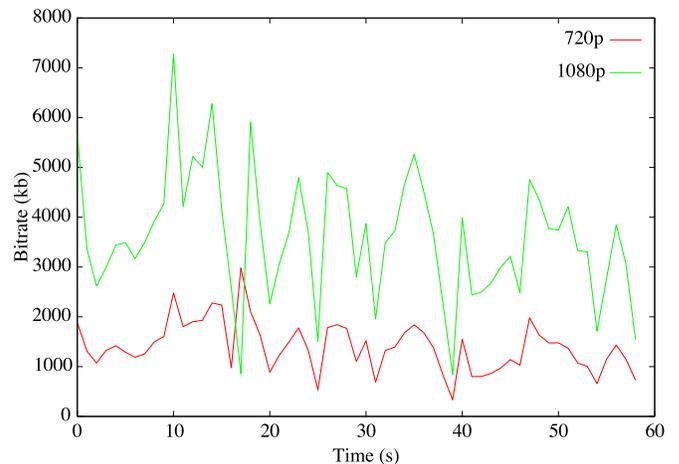


Fig. 5. Instantaneous bitrate for “Sintel”.

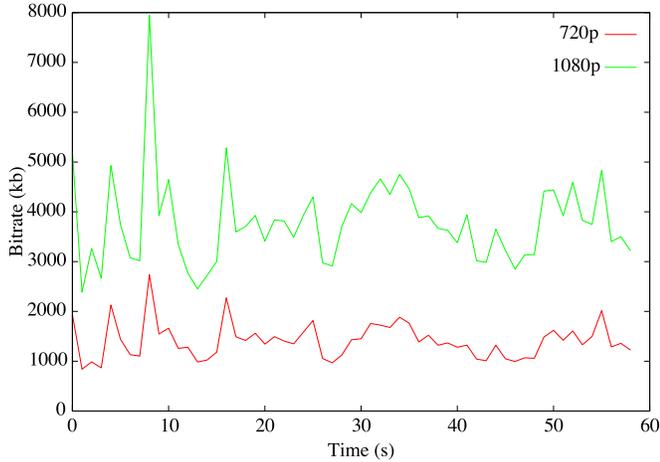


Fig. 6. Instantaneous bitrate for “Tears of Steel”.

Regarding the test-setup, the evaluation is done through simulations. First, we create and encode the videos and extract a trace of their frames (type of frames, size, transmission time...). Then we use the algorithm defined in the RFC of FLUTE to packetize the video in blocks and symbols, and then we calculate the number of symbols after applying FEC. Specifically, the following code rates are taken into account for the measures: 0.3 (233% parity), 0.5 (100% parity), 0.8 (25% parity) and 1 (no parity). Obviously, the more the protection, the more the bandwidth required. Also, simulations apply ideal decoding: in order to recover a block, it is only necessary to receive as many encoding symbols as the number of source symbols in that particular block.

With regards to the transmission, we simulate channel losses using the two-state Markov model (also known as Gilbert model) [16]. This model is widely used in the literature, since it simulates well the burst losses, typical in wireless networks. There are three different packet loss rates regarded in the study: 5%, 25% and 50%. In reception we check for every frame if enough packets have been received to rebuild each frame with a certain quality. If not the case, these packets are requested to the DASH server. Then, the video is rebuilt, and the PSNR is calculated by comparing the rebuilt video with the original one.

In order to present an appropriate evaluation in mobile wireless networks, all the simulations have been carried out in a controlled environment, where packet losses are caused by both node mobility and channel interferences. Also, it should be noted that the formats and protocols used to carry out the measures (H.264 AVC, DASH/HTTP, FLUTE/UDP, MPD) permit to deliver content in unicast and broadcast environments and they

guarantee accessibility over a wide range of networks. Also we have used formats applicable to variable loss channels, which adapt to changing conditions using FEC without degrading excessively the Quality of Experience.

Finally, it should be mentioned that, for the sake of clarity, it is considered that initially all clients are connected to the FLUTE channel that best fits their channel losses and clients do not switch among channels. Also, for simplicity we have considered that the delay of the network as well as the overhead of the underlying protocols of DASH is insignificant.

6. RESULTS AND ANALYSIS

6.1 Introduction

This section presents the evaluation of the system proposed. The different studies consider the two videos aforementioned.

First, the multicast transmission through FLUTE is analyzed. Specifically, Section 6.2 evaluates both the PSNR and the bandwidth for different configurations of the FEC protection and different losses. The PSNR obtained is a good indicator to evaluate the subjective quality of a video.

In this sense, Section 6.3 establishes a relation between the percentage of data downloaded and the PSNR. Receiving a certain percentage of packets guarantees a good level of PSNR. If the amount of packets received by means of FLUTE is not enough, clients are able to request for lost segments to the DASH repair service.

Finally, Section 6.4 compares in terms of bandwidth our proposal with a unicast system based on DASH.

6.2 Evaluation of the bandwidth

Fig. 7 and Fig. 8 show the values of PSNR and bandwidth consumed for different FEC configurations after a FLUTE transmission of the video “Sintel”, for 720p and 1080p respectively. Each vertical bar displays the bandwidth used by a specific AL-FEC configuration and contains three points that represent the values of PSNR for each percentage of losses. Above each bar, the values in brackets indicate the code rate applied to I, P and B-frames, in this order. In all cases shown, I-frames have at least the same protection as the P and B-frames have. Results show the quantitative effect of increasing the protection for the different types of frames. This way, increasing the FEC of I-frames improves considerably the PSNR. In contrast, increasing

the FEC of B-frames hardly improves the PSNR and moreover increases the bandwidth. Therefore, the use of a lot of protection is not always a good solution in terms of bandwidth. As figures show, there is a trade-off between the PSNR obtained and the bandwidth.

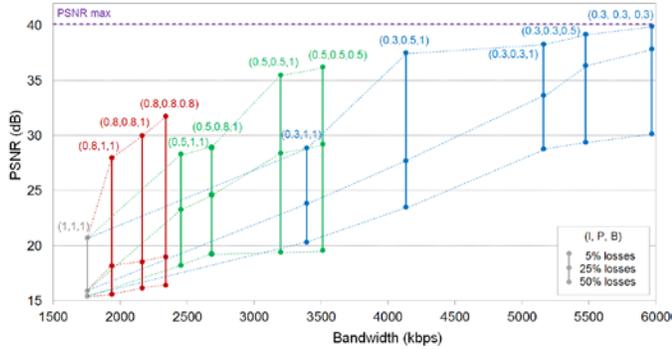


Fig. 7. PSNR and bandwidth evaluation in Video 1: “Sintel”, 720p.

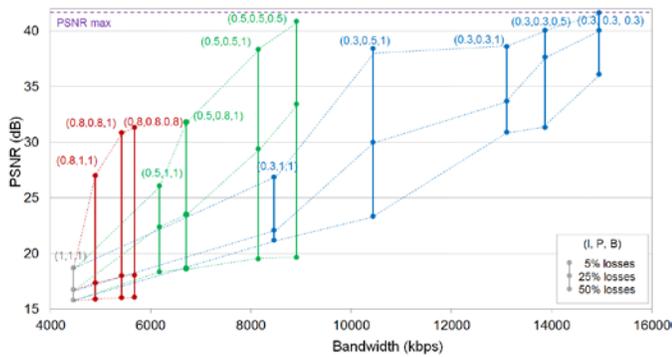


Fig. 8. PSNR and bandwidth evaluation in Video 2: “Sintel”, 1080p.

Comparing both figures, the behavior of the PSNR is similar when the FEC increases, providing similar values of PSNR for the same FEC protection. However, the bandwidth of Fig. 8 is rather higher than the bandwidth obtained in Fig. 7, due to the different encoding rate (as shown in Table I).

This way, using the information shown in Fig. 7 and Fig. 8, it is possible to configure the video transmission according to the available bandwidth and the desirable quality. In this sense, Table II shows the relation between the PSNR and the MOS (Mean Opinion Score) [17]. MOS is a method commonly used to quantify the video quality based on the human quality impression.

TABLE II
PSNR TO MOS CONVERSION

PSNR (dB)	MOS
> 37	5 (Excellent)
31 - 37	4 (Good)
25 - 31	3 (Fair)
20 - 25	2 (Poor)
< 20	1 (Bad)

By fixing a level of MOS and a maximum bandwidth it is possible to choose the optimal FEC configuration for a certain percentage of losses. For instance, in Fig. 8, if a server wants to offer an “excellent” service (MOS=5) to clients with 5% of losses with the minimum bandwidth, the server will use a FEC configuration of (0.5, 0.5, 1). Note that the configuration of (0.3, 0.3, 1) only increases the PSNR 0.27 dB at the expense of increasing the bandwidth a 61%.

When losses are very high (50%) it is needed to use a high protection to obtain a high level of MOS, thus increasing considerably the bandwidth. Thus, in Fig. 7, to get a “fair” quality (MOS=3) it is necessary at least to protect the I and P-frames with a code rate of 0.3 (233% parity), consuming a bandwidth higher than 10000 kbps.

The analysis of the video “Tears of Steel” offers the same conclusions. Fig. 9 and Fig. 10 present the results obtained for qualities 720p and 1080p, respectively.

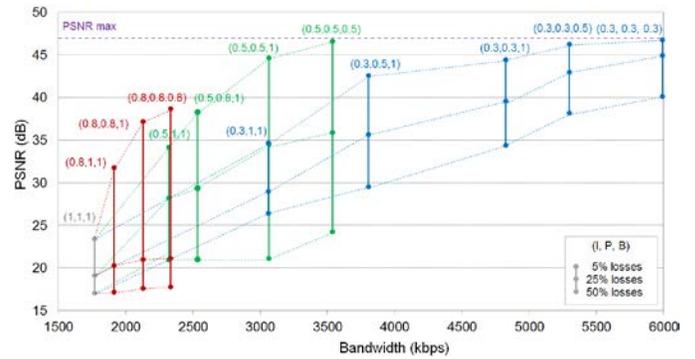


Fig. 9. PSNR and bandwidth evaluation in Video 3: “Tears of Steel”, 720p.

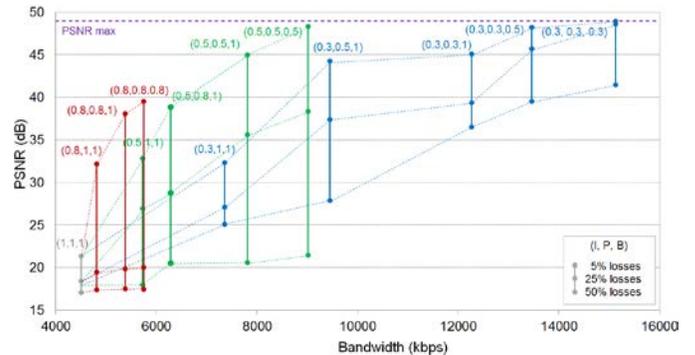


Fig. 10. PSNR and bandwidth evaluation in Video 4: “Tears of Steel”, 1080p.

In this case, the maximum PSNR of both videos is higher than the one obtained in Fig. 7 and Fig. 8. As mentioned, it is very important to choose the best configuration for each case. As a good example, in Fig. 10 the FEC configuration of (0.3, 1, 1) needs similar

bandwidth that the configuration (0.5, 0.5, 1) but provides values of PSNR much worse (34.63 dB against 44.64 dB for 5% losses, and 28.99 dB against 34.51 dB for 25% losses).

6.3 Percentage of data downloaded

Fig. 11 and Fig. 12 depict the relationship between the percentage of data downloaded through FLUTE and the PSNR of the reconstructed video for the video “Sintel”. Both parameters depend on the channel losses, the bandwidth used (determined by the amount of FEC applied), and the configuration of the FEC block. Note that figures consider those FEC configurations shown in previous studies (the FEC protection for each point in the figures is not shown for the sake of clarity). As figures show, there is a minimum percentage of data needed to obtain a good level of PSNR. In general, it is needed to receive at least 90% of bits to obtain an acceptable PSNR. If a client receives less than 90% of bits the use of a repair DASH service is needed. Clients with more than 90% of bits downloaded can decide to connect to the DASH service to obtain bits lost in order to improve their PSNR. As Fig. 11 and Fig. 12 show it is needed to receive more than 95% of bits to obtain an excellent MOS.

On the other hand, with FLUTE it is possible to obtain a good video quality if an appropriate protection is applied, even in channels with high losses, as Fig. 11 and Fig. 12 show. Nevertheless, having a high percentage of bits received does not guarantee a good PSNR. For instance, this is the case of receiving the major part of P and B-frames but losing some I-frames. As long as a minimum percentage of bits have been received, it is more important which packets are received than how many.

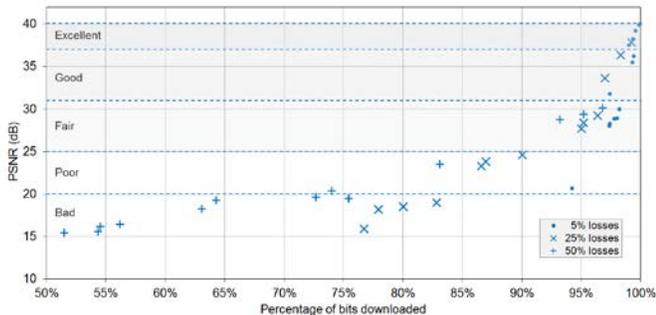


Fig. 11. Relationship between the percentage of bits downloaded and the PSNR (Video 1).

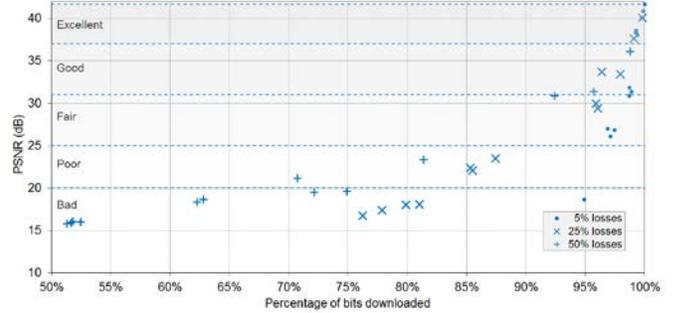


Fig. 12. Relationship between the percentage of bits downloaded and the PSNR (Video 2).

The results obtained with the video “Tears of Steel” are very similar, as Fig. 13 and Fig. 14 show.

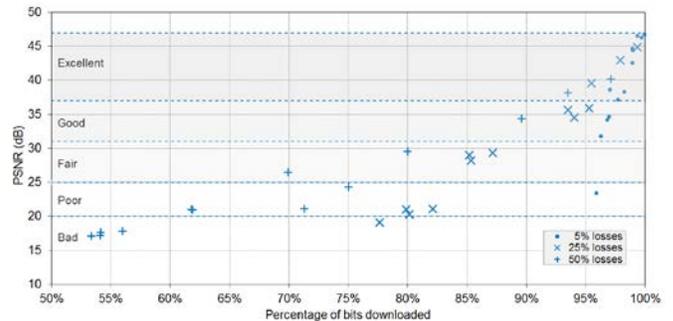


Fig. 13. Relationship between the percentage of bits downloaded and the PSNR (Video 3).

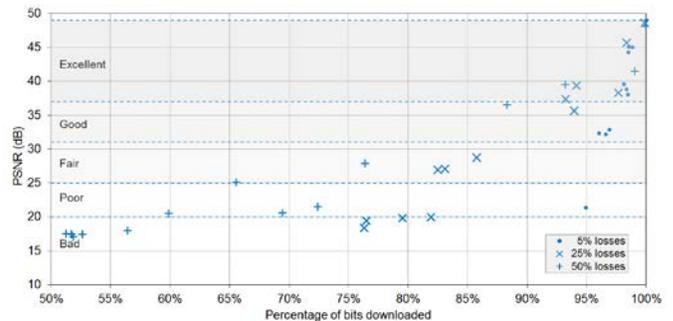


Fig. 14. Relationship between the percentage of bits downloaded and the PSNR (Video 4).

6.4 Evaluation of FLUTE/DASH traffic

Previous studies have analyzed the use of FLUTE multicast to stream video. This section considers the use of DASH to repair data and compares the proposed solution with a unicast transmission service based only on DASH. In our proposal, all clients share the available bandwidth and therefore the amount of data saved by using the multicast channel depends on the number of users in the cell. Similarly, the overall amount of data actually delivered by the unicast channel depends on the

packet loss in the channel between the server and the clients.

As explained in Section 4, the server sends the same video with two different qualities, by using two different FLUTE channels. Each FLUTE channel has an associated DASH server, which is used to repair missing data. Clients that detect high losses are intended to connect to the channel with lower quality (where the FEC protection is higher) whereas clients with low losses can connect to the channel with more quality.

As a particular scenario, we suppose that a server wants to offer a “good” MOS (MOS=4), that is, a PSNR higher than 31 dB. To that extent the server will use a FEC protection that consumes the minimal bandwidth and that it provides a PSNR higher than 31 dB. For the video at 720p it will be considered the case of 25% losses and for the video at 1080p it will be considered the case of 5% losses. As we have commented, videos at 720p need more FEC protection than videos at 1080p, since the users who consume the video at 720p have more channel losses.

In this way, Fig. 7 shows that for the video “Sintel” the best FEC configuration at 720p is (0.3, 0.3, 1), with a bandwidth of 5160 kbps; whereas with 1080p (Fig. 8) the best configuration is (0.8, 0.8, 0.8), with a bandwidth of 5420 kbps. Apart from the bandwidth consumed by the FLUTE server, the total bandwidth depends on the bandwidth used by the DASH servers. Clients will request lost frames depending on the losses perceived. Thus, if the video is not protected enough, the amount of DASH requests will be high.

Fig. 15 shows a comparison between the system proposed and a system based only on DASH, that is, users requesting video packets with a certain quality. Figure depicts that the solution based on FLUTE/DASH reduces considerably the bandwidth as the number of clients increases. Specifically, the bandwidth obtained using FLUTE/DASH for the video at 720p is lower than the one obtained with unicast when there are more than 4 clients in the system. At 1080p, only 2 clients are necessary to obtain better results with FLUTE/DASH comparing with only DASH. On the other hand, comparing the two resolutions in the FLUTE/DASH system, we can see that initially the video at 1080p consumes more bandwidth than the video at 720p and when the number of clients increases the trend is reversed.

Furthermore, Fig. 15 allows to calculate easily the bandwidth used for a certain number of users. For example, in a scenario where 6 clients have low losses and 4 have high losses, the total bandwidth consumed is

the sum of the bandwidth for the case of 1080p for 6 users and the bandwidth for the case of for 4 users, in the example 11.80 Mb using FLUTE/DASH and 30 Mb using only DASH.

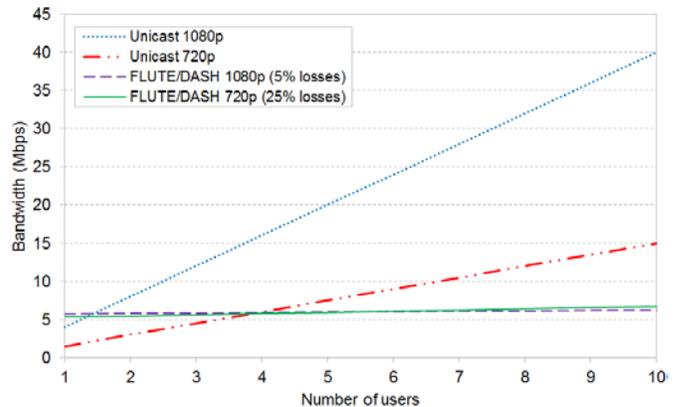


Fig. 15. Comparison among the proposed solution and unicast with video “Sintel”. FEC configuration: FLUTE/DASH at 720p (0.3, 0.3, 1), FLUTE/DASH at 1080p (0.8, 0.8, 0.8).

Regarding the video “Tears of Steel”, as Fig. 9 and Fig. 10 depict, the best configuration at 1080p to achieve the “good” MOS level with the minimum bandwidth is (0.8, 1, 1), whereas at 720p the best configuration is (0.5, 0.5, 1). The results are shown in Fig. 16, providing similar conclusions: the reduction of the bandwidth using the proposed FLUTE/DASH system increases proportionally to the number of users. On the other hand, although it is not shown in the graph, the bandwidth consumed by FLUTE/DASH at 720p overcomes the bandwidth consumed by FLUTE/DASH at 1080p when there are more than 45 users, due to the increase of DASH requests by the users that have high losses (which are connected to the channel that sends the video at 720p).

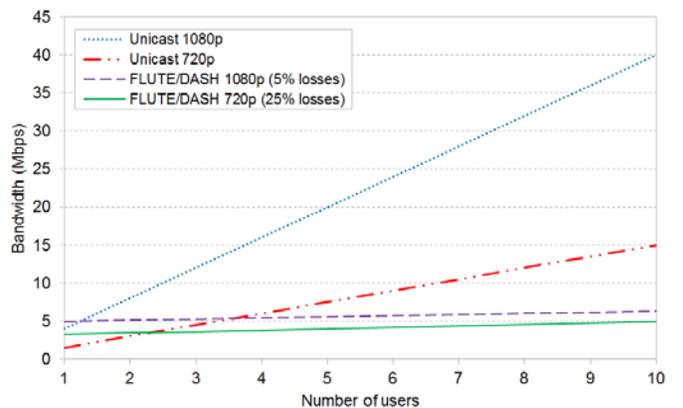


Fig. 16. Comparison among the proposed solution and unicast with video “Tears of Steel”. FEC configuration: FLUTE/DASH at 720p (0.5, 0.5, 1), FLUTE/DASH at 1080p (0.8, 1, 1).

As another example, Table III shows the average amount of data requested to the DASH server as a percentage of the video size, for the videos considered in this paper, for different levels of MOS and in three different scenarios. Specifically, two levels of MOS are analyzed: a “fair” level and a “good” level. In order to achieve these levels, the server chooses the optimal FEC configuration. Thus, for the channel which sends the video at 1080p, the server considers the FEC configuration that minimizes the bandwidth and that provides values of PSNR higher than 25 dB (in the “fair” level) and 31 dB (in the “good” level) for 5% channel losses. The same condition is applied for the video at 720p but considering 50% channel losses. Clients with 25% losses also will connect to the channel which sends at 720p. Fig. 7-Fig. 10 are useful to find the optimal FEC values, which are shown in Table IV.

Table III uses the optimal FEC configuration depicted in Table IV for every case to calculate the efficiency of each configuration, expressed as the bandwidth improvement with respect to unicast. As shown in Table III, for the three scenarios, two videos and two qualities are considered, the hybrid FLUTE/DASH architecture proposed reduces the network traffic over 90% when there are 100 clients in the mobile network. Since most of the data is downloaded through FLUTE, the bandwidth is used in a more efficient manner. Regarding the DASH recovery, the percentage of data requested to the DASH server is lower as the MOS level improves, since it is necessary to increase the amount of FEC in order to achieve a higher MOS level (as the FEC protection increases the losses and thus the number of DASH requests decreases). As expected, when the DASH recovery decreases the efficiency improves.

TABLE III
IMPROVEMENTS OF HYBRID FLUTE/DASH WITH RESPECT TO UNICAST

MOS Level	Sintel		Tears of Steel	
	3 (Fair)	4 (Good)	3 (Fair)	4 (Good)
Scenario 1:				
34 users with 5% packet loss, 33 with 25% and 33 with 50%				
DASH recovery	1.9%	0.6%	8.5%	2.6%
Efficiency	92.0%	93.7%	85.4%	91.0%
Scenario 2:				
80 users with 5% packet loss, 10 with 25% and 10 with 50%				
DASH recovery	2.4%	0.7%	4.6%	2.8%
Efficiency	94.1%	96.2%	92.4%	93.6%
Scenario 3:				
10 users with 5% packet loss, 10 with 25% and 80 with 50%				
DASH recovery	2.0%	0.8%	13.2%	3.1%
Efficiency	89.0%	92.1%	72.1%	86.4%

TABLE IV
OPTIMAL FEC CONFIGURATION FOR DIFFERENT QUALITIES AND LEVELS OF MOS

Quality	Sintel		Tears of Steel	
	3 (Fair)	4 (Good)	3 (Fair)	4 (Good)
720p	(0.3,0.3,1)	(0.3,0.3,0.3)	(0.3,1,1)	(0.3,0.3,1)
1080p	(0.8,1,1)	(0.8,0.8,0.8)	(0.8,1,1)	(0.8,1,1)

Finally, it can be noted that the solution hereby presented is based on video streaming. The case of video on demand is less demanding, since the requirements of bandwidth are not so strict, and there are other alternatives that can be used. One of them is to consider continuous retransmissions. This alternative is similar to that explained before, but using FLUTE carousels instead of DASH repair sessions to recover the packets previously lost. This solution has been presented by the authors in different papers, such as [11], which shows that there is an optimum configuration of AL-FEC codes, depending on the channel losses and other characteristics of the files sent, that minimizes the download time of a multimedia content.

7 CONCLUSIONS AND FUTURE WORK

In this paper we have proposed the use of a hybrid broadcast/unicast architecture based on FLUTE and DASH to deliver video. As results have shown, this proposal can help reduce considerably the bandwidth (and thus the network traffic) due to the use of FLUTE multicast, without degrading the video quality. To that extent, our proposal uses AL-FEC mechanisms to correct data and DASH to repair lost frames. Specifically, an appropriate configuration of UEP for different frames allows to maximize the bandwidth utilization of the mobile network. This optimal configuration depends on the available bandwidth of the transmission channel and the MOS level that the server wants to offer. Specifically, results have shown that for the different scenarios considered, our proposal can reduce the bandwidth a 90% compared with a unicast solution.

In this sense, it is possible to group different frames with small sizes (P and B-frames) within the same FLUTE block in order to apply jointly protection, which would improve both the PSNR and the bandwidth utilization. This jointly coding as well as an exhaustive analysis of the size of the buffers in reception and the delay of the network is part of the future work. Finally, it is intended to carry out a complete subjective QoE evaluation in a real environment, using mobile devices in a wireless network. In this sense, when evaluating the

system in a real mobile environment, there are different parameters to consider [3], for instance the use of protocols and formats that allow to send content in hybrid (multicast/unicast) networks, or formats which adapt to changing environments, without degrading excessively the Quality of Experience.

ACKNOWLEDGMENTS

This work was supported by the 11012 ICARE (Innovative Cloud Architecture for Real Entertainment) project within the ITEA 2 Call 6 Program of the European Union.

REFERENCES

1. ETSI TS 126 346 v11.3.0. Universal Mobile Telecommunications Systems (UMTS); LTE; Multimedia Broadcast/Multicast Service (MBMS); Protocols and Codecs, 2013.
2. Lecompte D, Gabin F. Evolved Multimedia Broadcast/Multicast Service (eMBMS) in LTE-Advanced: Overview and Rel-11 Enhancements. *IEEE Communications Magazine* 2012; 50(11): 68-74.
3. Stockhammer T, Luby MG. DASH in mobile networks and services. *IEEE Visual Communications and Image Processing (VCIP)*, San Diego, CA, 2012.
4. Seeling P, Reisslen M. Video transport evaluation with H.264 video traces. *IEEE Communications Surveys & Tutorials* 2012; 14(4): 1142-1165.
5. Zhao S, Tuninetti D, Ansari R, Schonfeld D. Multiple description coding over multiple correlated erasure channels. *Transactions on Emerging Telecommunications Technologies* 2012; 23(6): 522-536.
6. Lin CH, Wang YC, Shieh CK, Hwang WS. An unequal error protection mechanism for video streaming over IEEE 802.11e WLANs. *Computer Networks* 2012; 56(11), 2590-2599.
7. Paila T, Walsh R, Luby M, Roca V, Lehtonen R. FLUTE – File delivery over unidirectional transport. *IETF RFC 6726*, 2012.
8. Luby M, Watson M, Vicisano L. Asynchronous layered coding (ALC) protocol instantiation. *IETF RFC 5775*, 2010.
9. Ameigeiras P, Ramos-Munoz JJ, Navarro-Ortiz J, Lopez-Soler JM. Analysis and modeling of YouTube traffic. *Transactions on Emerging Telecommunications Technologies* 2012; 23(4), 360–377.
10. ISO/IEC 23009-1. Dynamic adaptive streaming over HTTP (DASH) – Part 1: media presentation description and segment formats, 2012.
11. de Fez I, Fraile F, Belda R, Guerri JC. Analysis and evaluation of adaptive LDPC AL-FEC codes for content download services. *IEEE Transactions on Multimedia* 2012; 14(3): 641-650.
12. Jenkac H, Stockhammer T, Xu W. Asynchronous and reliable on-demand media broadcast. *IEEE Network* 2006; 20(2): 14-20.
13. Neumann C, Roca V. Scalable video streaming over ALC (SVSoA): a solution for the large scale multicast distribution of videos. *1st Int. Workshop on SMDI*, Athens, Greece, 2004.
14. Lederer S, Müller C, Timmerer C. Dynamic Adaptive Streaming over HTTP Dataset. Proc. of the ACM Conference on Multimedia Systems (MMSys), Chapel Hill, North Carolina, 2012; 89-94.
15. Blender Foundation webpage, available at: <http://www.blender.org/blenderorg/>, accessed Sep. 2013.
16. Bai H, Atiquzzaman M. Error modeling schemes for fading channels in wireless communications: A survey, *IEEE Communications Surveys and Tutorials* 2003; 5(2): 2-9.
17. Ohm JR, *Multimedia communication technology: representation, transmission and identification of multimedia signals*. Springer Berlin Heidelberg, New York, 2004.