Analysis of Coding Strategies Within File Delivery Protocol Framework for HbbTV Based Push-VoD Services Over DVB Networks

Ferdaouss Mattoussi, Matthieu Crussière, Jean-Francois Helard, Gheorghe I. Zaharia

Analysis of Coding Strategies within File Delivery Protocol Framework for HbbTV based Push-VoD Services over DVB Networks

Ferdaouss Mattoussi, Matthieu Crussière, Jean-François Hélard, Gheorghe Zaharia
Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, F-35000 Rennes, France
{ferdaouss.mattoussi, matthieu.crussiere, jean-francois.helard, gheorghe.zaharia}@insa-rennes.fr

***Abstract*—Hybrid broadcast broadband TV (HbbTV) is a technique providing Push-VoD services over an interactive hybrid TV. These services are broadcast using File Delivery Protocol (FDP) which includes different coding strategies to ensure a reliable delivery. This protocol is characterized by three levels of data representation giving rise to segmentation of packet losses which may result in poor recovery capabilities. This paper provides a first thorough investigation of coding FDP framework for a reliable delivery of Push-VoD service over DVB networks. We propose Markov modeling for characterizing inter-layers loss propagation within FDP on a wide variety of burst erasure channels. Based on this analytical analysis and a simulation study, we determine the possible recovering areas and the accurate loss measurements within FDP. The latter are then used to effectively investigate and configure the different coding strategies provided within FDP. In addition, we present a suitable recovering strategy for FDP which guarantees transmission robustness against the broadcast network impairments.**

Keywords: HbbTV, Push-VoD services, AL-FEC, repetition code, Markov chain, DVB.

I. Introduction

A. The context of non-linear data transmission using HbbTV

The Television (TV) is probably the most cost-effective platform for informing, educating and entertaining people over the world. From the late 19^{th} century, the invention of the television was the work of many individuals and corporations in various parts of the world to deliver a device that supersedes previous technology and fills the people requirements. This induces the transition from black-and-white TV to the Digital TV (DTV) with high definition (HD) screens. The launch of the DTV and the big growth of the internet, becoming present to everywhere and everybody, have allowed the wealth of audio-visual experiences and the opportunity for apparition of the interactive TVs (iTV)s.
Connected TVs, known nowadays by SmartTV or NetTV, represent a category of iTVs capable of displaying, separately, both TV programs from a broadcasting network and additional services delivered via an Internet connection (e.g. catch-up TV services). The separation between these services and linear broadcast TV services makes the surfing between the two worlds inflexible and inappropriate. In order to cope with these inconveniences, many broadcasters are advocating solutions that seek to combine the functionalities of both broadcast and broadband tuners in a device to offer interactive applications with a seamless experience to the consumer. Several approaches have been proposed, in different countries, to achieve this requirement yielding an optimum experience (greatly enhance the viewing experience) such as DVB Globally Executable MHP (GEM) in Italy [1], Youview in UK, and Hybrid broadcast broadband TV (HbbTV) in Germany, France and elsewhere [2].

HbbTV technology was developed by the HbbTV group including both television broadcasters and CE companies and standardized by ETSI as a norm (ETSI TS 102 796) [2]. This technology is a pan-European initiative aiming at harmonizing video broadcast and high-speed transmission standards to entertainment programs broadcast on connected TVs, set-top-boxes (STB) and multi-screen viewing devices. HbbTV provides viewers a richer experience of enhanced and interactive hybrid TV, through the provision of innovative services on both broadcast and broadband networks, allowing the viewer to interact with the current TV program. More precisely, with such a system, "red button" services linked to TV programs is implemented to provide interactive applications in a much more functional and flexible way. In addition, HTML overlays are possible on full-screen TV pictures as well as the integration of a scaled TV image into a full-screen application: obtain real "hybrid" approach to "connected TVs".
Generally, there are two types of applications when referring to HbbTV; bound application or "broadcast-related" and unbound application or "broadcast-independent". The broadcast-related applications are tied to a current broadcast service (e.g., vote, quiz) whereas the broadcast-independent applications have no relation to any broadcast service and are accessible through a portal delivered by TV manufacturers (e.g., electronic program guides (EPGs), YouTube).
HbbTV represents a more specific profile of available tech-

This work has received a French state support granted to the AdicTV project through the 20th FUI (transverse inter-ministry funding) program. The authors would also like to thank the "Image&Réseaux" and "Cap Digital" french business clusters for their support of this work.

nologies than a completely new technical development. It introduces only few new technical components. It is mainly based on DVB, CE-HTML and Open IPTV Forum (OIPF)s browser profile standards. The two latter standards define the main functionality of the HbbTV browser running on a TV, combining the TV picture with HTML pages [3]. DVB standard specifies and grants more DVB-related integration capability that is required to signaling and carriage of the interactive applications (HbbTV applications) in the hybrid environments [4].

HbbTV combines data and applications received via the broadcast signal with services, applications and contents provided via the Internet. HbbTV applications are signaled in Application Information Table (AIT) in Program Specific Information/Service Information (PSI/SI) tables. Depending on the HbbTV application type, it is possible to deliver HbbTV applications in two ways [2]:

- The first consists in encapsulating application data into Digital Storage Media - Command and Control (DSM-CC)[1] object carousels and sending it by broadcast channel. The application, which is fragmented (into separate files or into separate updates to files or trees), can be delivered synchronously with the program signal and presented progressively to the user with a different carousel period, depending on the nature and importance of each fragment. This allows time accurate data such as questions or answers for an interactive quiz show to be conveyed. This protocol is specified to transfer application data[2] and stream events (data streams) [4].
- The second way consists in signaling application as a link to remote HTTP server over broadcast channel in AIT in PSI/SI tables and download applications over broadband channel. The data from both delivery channels are processed seamlessly in one application.

Since the appearance of HbbTV, several releases have been made proposing and/or adding services, features and enhancements. For instance, the most important advancement of HbbTV v2.0 version standardized in May 2015, was the integration of selections from HTML 5 and other next generation web technologies, the support of second-screen devices allowing a platform-independent user experience, as well as the addition of the Push-VoD service [5]. The present paper focuses on this release and more precisely on the Push-VoD service.

[1]DSM-CC is used to deliver Internet data via the broadcast channel. The DSM-CC allows to transmit small applications (like teletext replacement) to viewers who do not connect their devices to the Internet, by minimizing web server load by transmitting a small autostart launcher application and link to web resources in a second usage step. However the conventional way of accessing HbbTV applications is by a HTTP request from a broadband-connected service.

[2]set of files comprising an application, including HTML, Javascript, CSS and non-streamed multimedia files

B. Reliability of unidirectional content delivery

One of the main requirements of the file broadcast service is the reliable delivery, which means efficiently dealing with the problems caused by erasure-prone broadcast networks. Over broadcast networks, both the absence of any return channel and the unlimited scalability in terms of number of receivers who behave in a completely asynchronous way, prevent the use of repeat request mechanism to adapt and individualize transmission rates according to some feedbacks sent by receivers. Therefore, content broadcasting heavily relies on Forward Erasure Correction (FEC) mechanisms to improve transmission robustness against channel erasures. The great advantage of using FEC with broadcast transmissions is that the same repair packet can recover different lost packets at different receivers.

In that perspective, upper layer FEC mechanisms can be very beneficial for file delivery/download services in unidirectional multicast/broadcast environments. It shortens the transmission duration and reduces bandwidth requirements while ensuring that the receivers reliably obtain the file. Application-Level FEC (AL-FEC) code, which is partially integrated in the application layer above the IP level, has become a key component of many content delivery systems. It is widely used as an efficient technique to increase the native reliability of the network to meet the requirements of the content download/delivery services through the recovery of packets lost in transmission, as it can decrease download times as well as network traffic, since it avoids the request of lost packets [6]. There are many studies concerning reliable file content delivery over unidirectional packet erasure channels. For instance, to increase the robustness of DVB-H file delivery in IP Datacast Content Delivery Protocol (IPDC CDP), a systematic Raptor AL-FEC code was adopted with the File Delivery over Unidirectional Transport (FLUTE)[3] protocol for use instead of the MPE-FEC of the link layer [7]. Similar to the DVB-H, the same Raptor AL-FEC code with FLUTE protocol was used in IPDC CDP for DVB-SH file delivery [8]. In DVB-IPTV Content Download Services (CDS), the same AL-FEC based Raptor code of the IPDC CDP file delivery over DVB-H and DVB-SH was selected with the FLUTE protocol [9]. Low Density Parity Check (LDPC)-Staircase code is the AL-FEC scheme being chosen along with FLUTE/Asynchronous Layered Coding (ALC) protocol to improve the reliability and efficiency of push video services in the Japanese ISDB-Tmm standard[4] for mobile multimedia [10].

C. Contributions

The question we address in this article is how File delivery protocol (FDP) manages for reliable HbbTV Push-VoD

[3]FLUTE, defined in RFC 3926, is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. Its main characteristics is that this protocol offers reliability in the transmission. DVB-H uses FLUTE to send the Electronic Service Guide (ESG).

[4]This is a broadcast technology (based on ISDB-T), used for digital terrestrial TV services and suited to mobile environments.

service delivery over broadcast networks. Our contribution is largely methodological contrary to the massive majority of literature studies, which have proposed and/or analyzed different error correction mechanisms within transport protocol under various assumptions or independently of protocol configuration and parameters.
The main contributions of this work are summarized as follows:

- First, this paper presents a reference work about the modeling and analysis of inter-layers loss propagation within FDP on a wide variety of burst erasure channels. The developed model enables to determine system performance metrics that helps to understand and effectively dimension different data recovery strategies within FDP. Our analytical model was briefly introduced in a preliminary study in [11] and is presented herein in all details with the complete set of mathematical derivations and performance analyses for various system settings.
- Second, using the provided analytical analysis on loss propagation within FDP, we define and establish the possible file recovering areas for the coding strategies provided within FDP.
- Third, we propose and investigate a novel solution based on LDPC-Staircase code for FDP protocol to increase the robustness of HbbTV based Push-VoD services delivery over burst-loss channels. This choice of AL-FEC code is based on many interesting key features of LDPC-Staircase as already discussed in [12] and shown in details in Section IV.
- On this basis, an in-depth analysis of optimal configuration of the coding FDP framework for a reliable VoD delivery over burst-loss channels is provided. Following the approach proposed in [12] and [13] a comparison between the provided recovery strategies within FDP is given to determine the best one that guarantees transmission robustness in front of correlated erasure channel impairments. In this paper, an in-depth investigation is carried out while taking into account the correction capacity and the amount of data required to be sent to receive a file correctly.

This paper is organized as follows. An introduction to the FDP protocol and the reliable delivery methods of the Push-VoD services is given in Section II. An investigation and measurement of the loss run distribution within the FDP protocol over a large variety of burst erasure channels is detailed in Section III. A description of different coding strategies and the possible recovering areas within FDP are explained in Section IV. An in-depth analysis of optimal configuration and a performance analysis of the recovery strategies within FDP for a reliable file delivery using FDP protocol is discussed in Section V. Finally, Section VI concludes this paper.

II. HbbTV based Push-VoD services

A. Definition

Push-VoD is an established approach to deliver interactive Video-On-Demand (VoD) (i.e., non real-time Audio/Video (A/V) content) services to viewers by providing videos having been previously broadcast and stored on their TV reception equipment. Push-VoD system uses a local storage device (e.g, Set-Top-Box (STB)) to host a wide range of video contents. It works so that the service provider initiates the download of content items from the broadcast to the local storage device automatically and without explicit user request. Once the content is completely downloaded it can be offered for playback to provide an instantaneous and error-free VoD experience. Following this concept, an HbbTV application schedules the download of movies or other audio-visual content prior to its presentation. Moreover, viewers enjoy the benefits of interactivity: they can select and watch whenever they want without waiting time, content that has already been downloaded. The downloaded content is usually deleted after a predefined period to free up space for new content. Push-VoD service is used to provide true VoD and to offload streaming requests to the VoD servers in the network for high demand blockbuster movies and to overcome the problem of broadband connection.

For HbbTV, this service is performed based on the so-called File Delivery Protocol (FDP) and Content Access Download Descriptor (CADD) protocol to broadcast and download the contents respectively [5]. Push-VoD provides the highest quality of VoD service even for contents of high data volume (HD, 3D, Ultra HD) and even if a high bandwidth broadband channel is not available [5].
The reliable broadcast of HbbTV based Push-VoD services using FDP protocol can be insured using three different methods. The repeat transmission technique of files is the first and simplest way to enable reliable HbbTV based Push-VoD services delivery. The second possible method is the use of a AL-FEC technique which is mathematically a more efficient method of providing repair data and yields better system performance. Finally, HbbTV also enables combining AL-FEC and repeat methods which results in a AL-FEC repeat transmission technique. In each of these three options, file segment interleaving can also be activated to improve robustness against packet losses.
Another optional way to achieve the reliability is to use the File segment recovery technique based on the recovery URL via broadband channel. This recovery URL, provided in the Initialization Message, indicates the location where the terminal may retrieve the File segments which have not been received successfully. This technique is based on the terminal-recovery server communication. The terminal shall only use this mechanism in cases where a subset of the File segments of a file has been successfully received, but this subset is not sufficient for the file to be completely reconstructed and there is no broadcast period (i.e., all

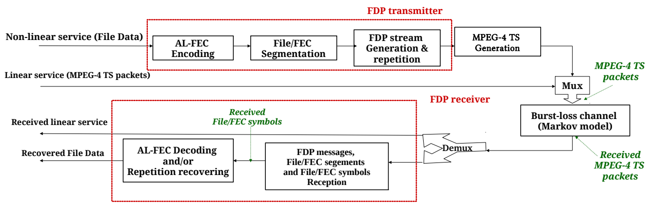


Fig. 1: Functional block for broadcasting HbbTV VoD service.

correction strategies via broadcast channel fail to recover the file) known to the terminal during which missing segments could be received [5].

B. FDP description

A VoD service may contain one or more A/V files. According to HbbTV 2.0 specifications, the delivery of each non real time A/V file content is assured using FDP. This protocol enables to generate an FDP stream consisting of FDP messages that end with a CRC to allow the terminal to check the integrity of the received message. Moreover, this stream consists of three categories of FDP messages:

- "Initialization messages": provide information regarding the file (e.g, file identification, file and/or FEC size, File and/or FEC segment size, file recovery URL, etc.) which is necessary for the terminal to initialize its reception.
- "Data messages": make up the A/V file content and may be sent in any order and each one may be sent more than once.
- "Termination messages": are sent after the last "Data message" to indicate to the terminal that this instance of the broadcasting of this file has ended.

All "Data messages" shall be sent between the first "Initialization message" and the first "Termination message".
Fig. 1 presents the functional block used to broadcast HbbTV VoD service using FDP. At the transmitter, each A/V file "File Data" undergoes different treatments such as AL-FEC encoding, segmentation, and repetition to generate an FDP stream. This process is more detailed in Fig. 2. If activated, an AL-FEC code is first applied on "File Data" to produce file "FEC Data" containing error correction data that the terminal can use to reconstruct the file in case it has not been properly received. "File Data" and "FEC Data" are both composed of symbols of equal size, $symb\_sz$. These files are then divided into segments to generate "File segments" and "FEC segments" respectively. "File segments" (resp. "FEC segments") shall be of equal size, $file\_seg\_sz$ (resp. FEC_seg_sz) except for the last "File segment" (resp. "FEC segment") which may be smaller.
For a given file, the number of "File segments" and "FEC segments", denoted by N_{file} and N_{fec} respectively, can be determined based on the "File Data" size, $file\_sz$, and "FEC data" size, FEC_sz by

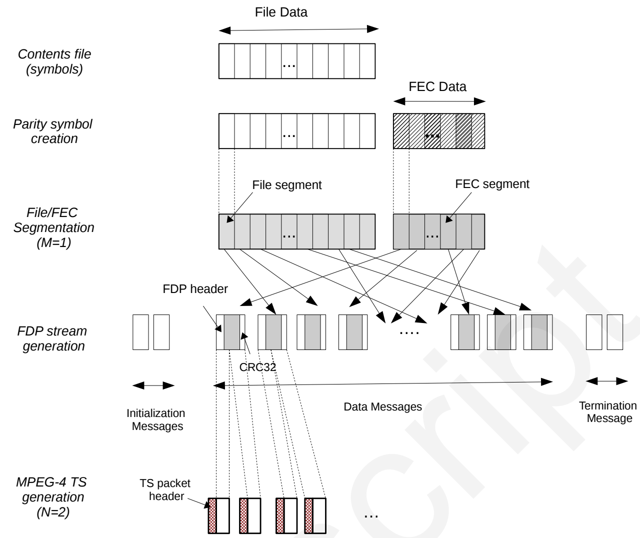


Fig. 2: Example of generation and insertion of the FDP messages in a DVB data pipe.

$$N_{file} = INT(\frac{file\_sz - 1}{file\_seg\_sz}) + 1, \quad (1)$$

$$N_{fec} = INT(\frac{FEC\_sz - 1}{FEC\_seg\_sz}) + 1. \quad (2)$$

Depending on $file\_seg\_sz$ and FEC_seg_sz, each File/FEC segment can contain more than one symbol. The number of symbols per segment, M, may be computed by

$$M = \frac{file\_seg\_sz \ (or \ FEC\_seg\_sz)}{symb\_sz} \quad (3)$$

In Fig. 2, each File (resp. FEC) segment contains one File (resp. FEC) symbol, i.e. $M = 1$.
After that each File and FEC segment will be carried in an FDP message of type "Data message". These "Data messages" may be sent in any order and each one may be sent more than once. Then, all FDP messages will be encapsulated within MPEG-4 TS packets according to the Data Piping model of the DVB data broadcasting specification [14], as shown in Fig. 2. Depending on FDP message size, FDP_msg_sz, each FDP message can be carried by one or more MPEG-4 TS packets. The number of MPEG-4 TS packets per FDP message, N can be computed as follows:

$$N = \frac{FDP\_msg\_sz}{MPEG{-}TS\_pckt\_pyld\_sz} \quad (4)$$

where FDP_msg_sz equals $file\_seg\_sz$ (or FEC_seg_sz) plus the header FDP size and CRC bytes. In Fig. 2, each FDP message is carried by two MPEG-4 TS packets, i.e. $N = 2$.
Moreover, the start of an FDP message may or may not be aligned with the start of the MPEG-4 TS packet payload. In this case, the $payload\_unit\_start\_indicator$ field of the TS packet header shall be used for FDP messages in the same way as specified for PSI sections in ISO/IEC 13818-1

[15].
As shown in Fig. 1, these MPEG-4 TS packets will be then multiplexed and broadcast with MPEG-4 TS packets carrying linear services. At the terminal, the reverse process will be performed.

III. Theoretical modeling of losses in FDP protocol

Over broadcast networks, MPEG-4 TS packets experience a correlated erasure channel. These packets contain FDP streams that are generated from FDP. Since the recovery strategies work at different levels within FDP (cf. Fig. 2), their behaviors depend on loss patterns appeared at a specific level within FDP rather than MPEG-4 TS packet loss distribution. Therefore, modeling and analysis of inter-layers loss spread within FDP on a wide variety of bursty erasure channels is mandatory to determine metrics allowing to measure the loss distortion that helps to understand and efficiently dimension the different recovery strategies within FDP.

In this section, based on a previous introduction of the subject in [11], we present a thorough analytical analysis of the loss propagation mechanisms within FDP. To that end, we set up a Markov chain which models the burst losses at various levels of the FDP protocol and whose parameters are derived to be directly expressed from the FDP and lossy channel parameters.

A. Burst-loss modeling

Let us model the loss distribution produced at each FDP system level by defining a Markov chain with appropriate transition probabilities matrix $\mathbf{P}$ and steady-state probabilities vector $\mathbf{\Pi}$. Fig.3 shows that FDP protocol concept can be represented by three levels where the service content changes shape and size passing from one level to another leading to have different loss dimensions within the protocol. More precisely, each level is characterized by a specific burst loss distribution as detailed and formalized in the following paragraphs.

1) Model at MPEG-4 TS packet level: Since in DVB networks there are as many channels as receivers with features that will also dynamically vary, we propose to use the two-state Markov model known as Gilbert model [16] to characterize the bursty losses over a wide variety of erasure channels. This choice is in line with many works that showed this model to be a good approximation to capture the temporal dependencies of packet losses [17]. As shown in Fig. 4, this model is characterized by packet transition probabilities (p_{pckt}, q_{pckt}) between the two states (no-loss (T) and loss (F)). p_{pckt} is the probability that the next packet is lost, provided the previous one has arrived. q_{pckt} is the opposite. If $p_{pckt} + q_{pckt} = 1$, the Gilbert model reduces to a Bernoulli model, i.e., erasure channel with Independent and Identically Distributed (IID) losses.
Since the Push-VOD service content is broadcast in MPEG-4 TS packet form, the burst loss distribution of these packets is modeled by the two-state Markov chain characterized by packet transition probabilities matrix $\mathbf{P_{pckt}}$ and packet stationary probabilities vector $\mathbf{\Pi_{pckt}}$. The $\mathbf{P_{pckt}}$ and the $\mathbf{\Pi_{pckt}}$ are determined as

$$\mathbf{P_{pckt}} = \begin{bmatrix} P(T/T) & P(F/T) \\ P(T/F) & P(F/F) \end{bmatrix} = \begin{bmatrix} 1-p_{pckt} & p_{pckt} \\ q_{pckt} & 1-q_{pckt} \end{bmatrix} \quad (5)$$

and

$$\mathbf{\Pi_{pckt}} = [\pi_{pckt}(T)\ \pi_{pckt}(F)], \quad (6)$$

where $\mathbf{\Pi_{pckt}}.\mathbf{P_{pckt}}=\mathbf{\Pi_{pckt}}$.
$\pi_{pckt}(T)$ and $\pi_{pckt}(F)$ represent respectively the packet stationary probabilities for state T and F and are computed by

$$\pi_{pckt}(T) = \frac{q_{pckt}}{p_{pckt}+q_{pckt}}$$
$$\pi_{pckt}(F) = 1-\pi_{pckt}(T) = \frac{p_{pckt}}{p_{pckt}+q_{pckt}}. \quad (7)$$

The average MPEG-4 TS packet loss rate (PLR) is given by

$$PLR = \pi_{pckt}(F). \quad (8)$$

The average burst-loss length (ABL) at MPEG-4 TS packet level can be given by [16]

$$ABL = \sum_{l=1}^{\infty} l.(1-q_{pckt})^{l-1}.q_{pckt} = \frac{1}{q_{pckt}}, \quad (9)$$

where l represents the length of the burst loss. ABL gives the expected mean loss run length.
The average inter-loss distance ($AILD$) at MPEG-4 TS

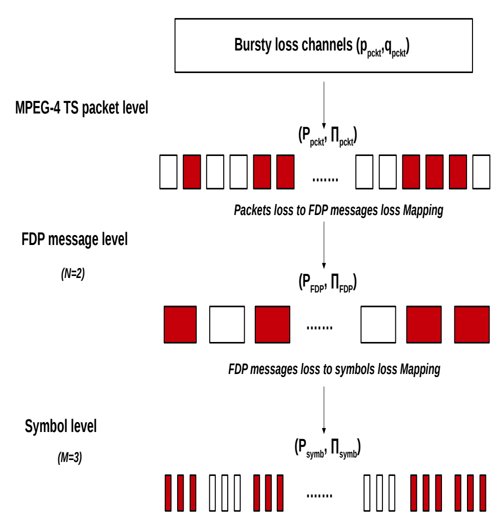


Fig. 3: The overall MPEG-4 TS packet to symbol loss mapping in FDP system. Red color represents the losses at each level.

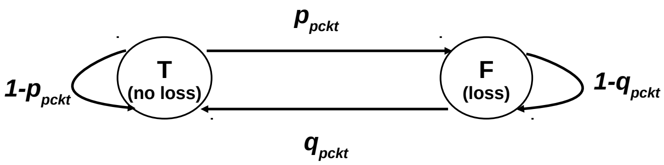


Fig. 4: Two-state Markov model.

packet level is given by [16]

$$AILD = \sum_{l=1}^{\infty} l.(1-p_{pckt})^{l-1}.p_{pckt} = \frac{1}{p_{pckt}}, \tag{10}$$

where l represents the length of burst no-loss. $AILD$ describes the average distance (spacing) between losses in terms of sequence numbers. It is useful to understand and foretell the spacing between loss events.

Consequently, Eq. (8) can be determined as follow

$$PLR = \frac{ABL}{ABL + AILD}. \tag{11}$$

2) Model at FDP message level: An FDP message has a no-loss state iff all the associated N MPEG-4 TS packets have no-loss states. This is defined by a single case denoted by T. An FDP message has a loss state iff at least one of the associated N MPEG-4 TS packets has loss state. There are $2^N - 1$ possible FDP loss states, depending on the MPEG-4 TS packet losses positions within the FDP message. These cases are denoted by F_i, $i \in \{0, \ldots, 2^N - 2\}$. Therefore, taking into account $\mathbf{\Pi_{pckt}}$ and $\mathbf{P_{pckt}}$, the loss distribution at FDP message level can be modeled by a Markov chain with two states (T and F=$\bigcup_{i=0}^{2^N-2} F_i$) and characterized by its transition probabilities matrix, $\mathbf{P^r_{FDP}}$, and its stationary probability vector, $\mathbf{\Pi^r_{FDP}}$. Matrix $\mathbf{P^r_{FDP}}$ is determined by

$$\begin{aligned}\mathbf{P^r_{FDP}} &= \begin{bmatrix} P_{FDP}(T/T) & P_{FDP}(F/T) \\ P_{FDP}(T/F) & P_{FDP}(F/F) \end{bmatrix} \\ &= \begin{bmatrix} 1-p_{FDP} & p_{FDP} \\ q_{FDP} & 1-q_{FDP} \end{bmatrix}, \end{aligned} \tag{12}$$

where

$$\begin{aligned} p_{FDP} &= 1-(1-p_{pckt})^N \text{ and} \\ q_{FDP} &= \frac{q_{pckt}.(1-p_{pckt})^{N-1}.[1-(1-p_{pckt})^N]}{p_{pckt}+q_{pckt}.[1-(1-p_{pckt})^{N-1}]}. \end{aligned} \tag{13}$$

Vector $\mathbf{\Pi^r_{FDP}}$ is given by

$$\mathbf{\Pi^r_{FDP}} = [\pi_{FDP}(T)\ \pi_{FDP}(F)], \tag{14}$$

where

$$\pi_{FDP}(T) = \frac{q_{pckt}.(1-p_{pckt})^{N-1}}{p_{pckt}+q_{pckt}} \tag{15}$$

$$\pi_{FDP}(F) = \frac{p_{pckt}+q_{pckt}.[1-(1-p_{pckt})^{N-1}]}{p_{pckt}+q_{pckt}}. \tag{16}$$

Details about probabilities computation can be found in Appendix A.

The average loss rate at FDP message level is equal to $\pi_{FDP}(F)$. The average burst loss length at FDP message level (ABL_{FDP}) is computed by

$$\begin{aligned} ABL_{FDP} &= \frac{1}{q_{FDP}} \\ &= \frac{p_{pckt}+q_{pckt}.[1-(1-p_{pckt})^{N-1}]}{q_{pckt}.(1-p_{pckt})^{N-1}.[1-(1-p_{pckt})^N]}. \end{aligned} \tag{17}$$

The average inter-loss distance at FDP message level ($AILD_{FDP}$) is computed by

$$AILD_{FDP} = \frac{1}{p_{FDP}} = \frac{1}{1-(1-p_{pckt})^N}. \tag{18}$$

3) Model at symbol level: As previously mentioned, an FDP message represents an FDP segment composed of M symbols with a header and CRC. Therefore, the loss of an FDP message leads to the loss of all the M associated symbols. The correct receipt of an FDP message leads to the correct receipt of all the M associated symbols. Therefore,

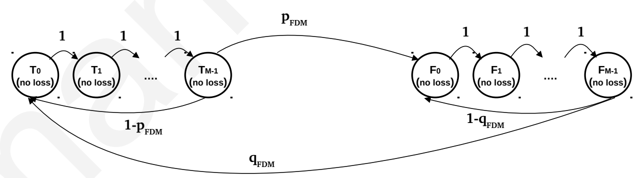


Fig. 5: The Markov model with $2M$ states for symbol loss distribution.

the symbol loss distribution can be modeled by a Markov chain with $2M$ states as shown in Fig. 5 and represented by a transition probabilities matrix $\mathbf{P_{symb}}$ as follows

$$\mathbf{P_{symb}} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 1 & 0 & 0 & \ddots & 0 \\ 1-p_{FDP} & 0 & 0 & \ddots & 0 & p_{FDP} & 0 & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 & 0 & 0 & \ddots \\ 0 & 0 & \ddots & \ddots & 0 & 0 & 1 & 0 & \ddots \\ 0 & 0 & \ddots & \ddots & 0 & 0 & 0 & \ddots & \ddots \\ 0 & 0 & \ddots & \ddots & 0 & 0 & 0 & \ddots & 1 \\ q_{FDP} & 0 & \ldots & \ldots & 0 & 1-q_{FDP} & 0 & \ddots & 0 \end{bmatrix} \tag{19}$$

The corresponding stationary probabilities vector $\mathbf{\Pi_{symb}}$ is equal to

$$\mathbf{\Pi_{symb}} = [\pi_{symb}(T_0)...\pi_{symb}(T_{M-1})\pi_{symb}(F_0)...\pi_{symb}(F_{M-1})], \tag{20}$$

with

$$\begin{cases} \pi_{FDP}(T) = \sum_{i=0}^{M-1} \pi_{symb}(T_i) \\ \pi_{symb}(T_0) = \pi_{symb}(T_1) = ... = \pi_{symb}(T). \end{cases} \quad (21)$$

Thus, we obtain:

$$\pi_{symb}(T) = \frac{1}{M}\pi_{FDP}(T). \quad (22)$$

Following the same approach, $\pi_{symb}(F)$ is computed as

$$\pi_{symb}(F) = \frac{1}{M}\pi_{FDP}(F). \quad (23)$$

We may note that the relationship between the FDP message level model and symbol level model is linear according to the M parameter.
Therefore, the loss rate at symbol level (SLR) which is equal to that of FDP message level, is computed by using Eq. (16) as

$$SLR = 1 - [(1 - PLR).(1 - \frac{1}{AILD})^{N-1}]. \quad (24)$$

From Eq.(17), the average burst-loss length at symbol level (ABL_{symb}) is computed as follows

$$ABL_{symb} = M.\frac{\frac{ABL}{AILD} + [1 - (1 - \frac{1}{AILD})^{N-1}]}{(1 - \frac{1}{AILD})^{N-1}.[1 - (1 - \frac{1}{AILD})^{N}]} \quad (25)$$

Finally, the average inter-loss distance at symbol level ($AILD_{symb}$) is obtained from Eq. (18) as

$$AILD_{symb} = M.\frac{1}{1 - (1 - \frac{1}{AILD})^{N}}. \quad (26)$$

B. Results and discussions

1) Models validation: In this section, we report some results obtained from numerical method to validate our Markov models proposed above. This validation is based on the comparison between SLR (resp. ABL_{FDP}) predicted by the proposed theoretical model using Eq. (24) (resp. using Eq. (17)) and the one estimated through a simulation study by measuring $\hat{p}_{FDP}$ and $\hat{q}_{FDP}$ parameters from the realistic loss distribution traces within received data sequences[5] at FDP message level [18]. These parameters are estimated as follows

$$\hat{p}_{FDP} = \frac{\sum_{i=1}^{n-1} m_i}{m_0}, \quad \hat{q}_{FDP} = \frac{\sum_{i=1}^{n-1} m_i}{\sum_{i=1}^{n-1} m_i.(i)}, \quad (27)$$

where, m_0 is the total number of no-loss FDP messages. m_i is the number of loss bursts having length i ($i = \overline{1, n-1}$ and $n-1$ is the longest burst loss length).

Thus, estimations of SLR, ABL_{FDP} and $AILD_{FDP}$ are calculated respectively as follows:

$$\widehat{SLR} = \frac{\hat{p}_{FDP}}{\hat{p}_{FDP} + \hat{q}_{FDP}}, \quad (28)$$

$$\widehat{ABL}_{FDP} = \frac{1}{\hat{q}_{FDP}}, \quad (29)$$

[5]For accurate estimation, we consider 10^9 sequences.

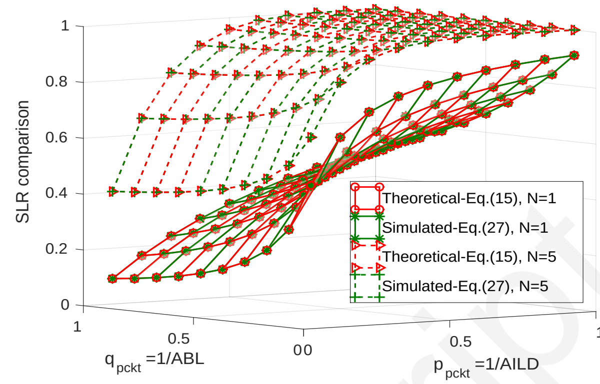


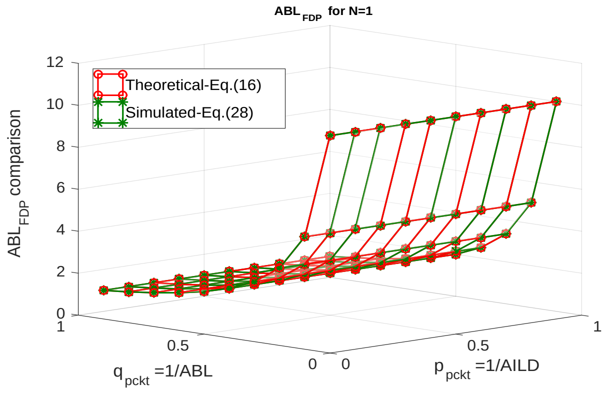
Fig. 6: Comparison of the theoretical and simulated values of the SLR for $N \in \{1, 5\}$.

$$\widehat{AILD}_{FDP} = \frac{1}{\hat{p}_{FDP}}. \quad (30)$$

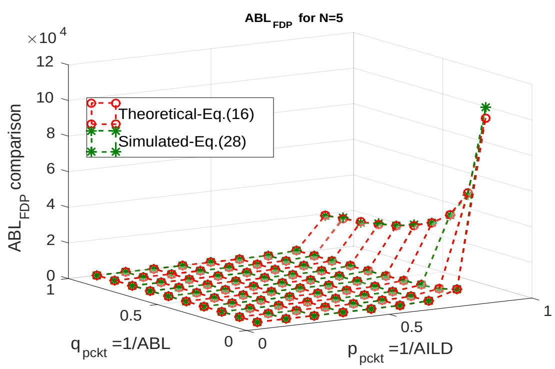
Fig. 6 (resp. Fig. 7 and 8) gives the SLR comparison (resp. ABL_{FDP} comparison and $AILD_{FDP}$ comparison) over different values of loss patterns at MPEG-4 TS packets (p_{pckt}, q_{pckt}) for $N \in \{1, 5\}$. These figures show that the simulation and the theoretical results match perfectly for all values of N and packet loss patterns. The proposed model is then valid at FDP level. Thereby, we can easily conclude it remains valid at symbol level, since we previously demonstrated there is a linear relationship between the two levels.

2) Loss propagation analysis: In this section, based on the proposed models, the loss spread within the FDP protocol is investigated. We determine the effect of the MPEG-4 TS packet loss distribution (measured by PLR, ABL, and $AILD$) and the FDP system parameters (N and M) on the loss distribution at symbol-level (measured by SLR, ABL_{symb} and $AILD_{symb}$). Theses symbol-level loss metrics enable to predict and understand the behavior of AL-FEC code and to select its good configuration as will be shown in Section V.
Fig. 9 depicts the SLR as a function of the PLR, according to Eq. 24, for $N \in \{1, 3, 5\}$ and $p_{pckt} \in \{0.1, 0.5\}$. As evident from Eq. 24 and the figure, the SLR linearly depends on the PLR according to a positive slope which is directly given by N and p_{pckt}. More precisely, when each FDP message contains one MPEG-TS packet ($N = 1$), the symbol-level loss rate is equal to MPEG-4 TS packet-level loss rate, i.e. $SLR = PLR$, regardless the average burst no-loss ($AILD$) and the average burst-loss (ABL) of MPEG-4 TS packets. On one other hand, for a fixed N, the larger the N and/or p_{pckt} (i.e. the lower the $AILD$) the higher the SLR. For instance, with $AILD$ = 10 (i.e., p_{pckt} = 0.1) and N = 5, the SLR is equal to 60% for PLR = 40% compared to 74% for PLR = 60%.

Fig. 10 provides the ABL_{symb} as a function of the ABL, according to Eq. (25), for $N \in \{1, 3, 5\}$ and $p_{pckt} \in \{0.1, 0.2, 0.5\}$ with $M = 1$. From this figure and



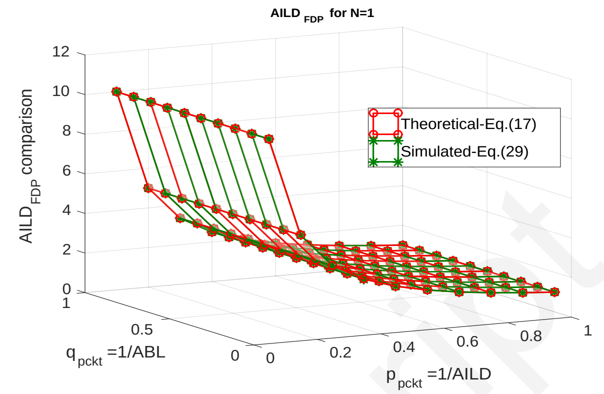
(a) ABL_{FDP} for N=1



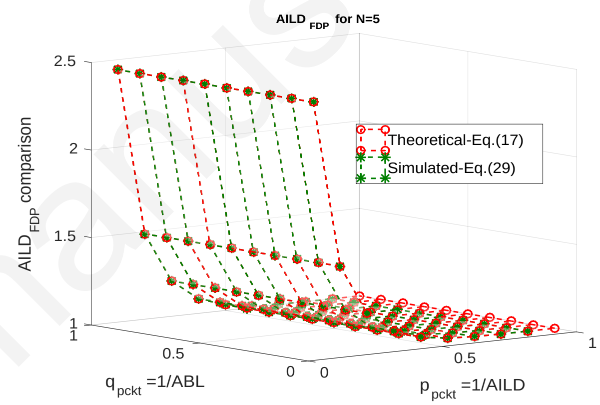
(b) ABL_{FDP} for N=5

Fig. 7: Comparison of the theoretical and simulated values of the ABL_{FDP}.

Eq. (25), we can draw three main conclusions. Firstly, when $N = 1$, the ABL_{symb} is directly proportional to the ABL independently of the $AILD$. In such case, the ABL_{symb} will grow according to coefficient M. This is due to the fact that when $M \neq 1$, the loss of a segment causes the loss of several consecutive symbols which translates into symbol-level bursts. Second, for a given value of $N > 1$, the ABL_{symb} varies linearly with the ABL and increases when the $AILD$ decreases (i.e. p_{pckt} increases). Finally, the effect on the ABL_{symb} of increasing system parameter N depends on both the ABL and the $AILD$. For example, for $AILD = 2$ ($p_{pckt} = 0.5$), the increase of N increases the ABL_{symb} for all ABL values. In contrast, for $AILD$ = 10 and $AILD$ = 5 (resp. p_{pckt} = 0.1 and p_{pckt} = 0.2), the increase of N induces different behaviors depending on the ABL value.
Fig. (11) gives the $AILD_{symb}$ as a function of the $AILD$, according to Eq. (26), for $N \in \{1, 3, 5\}$ with $M = 1$. From Eq. (26), it is reminded that the $AILD_{symb}$ only depends on the $AILD$ (i.e., p_{pckt}) and N parameter. Again,



(a) $AILD_{FDP}$ for N=1



(b) $AILD_{FDP}$ for N=5

Fig. 8: Comparison of the theoretical and simulated values of the ABL_{FDP}.

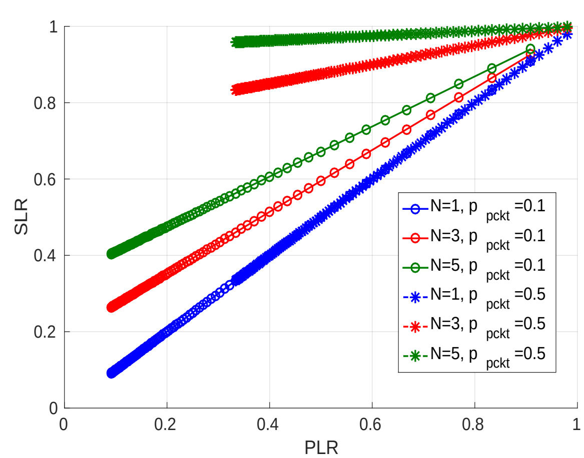


Fig. 9: SLR versus variation of PLR for $p_{pckt} \in \{0.1, 0.5\}$ and $N \in \{1, 3, 5\}$. $p_{pckt} = \frac{1}{AILD}$.

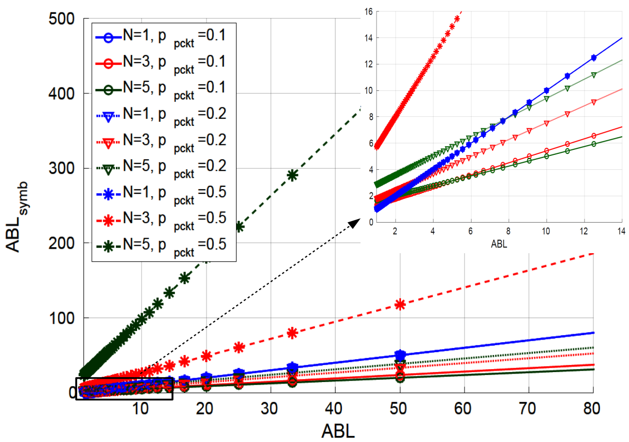


Fig. 10: ABL_{symb} versus ABL for different values of N and M = 1. $p_{pckt} = \frac{1}{AILD}$

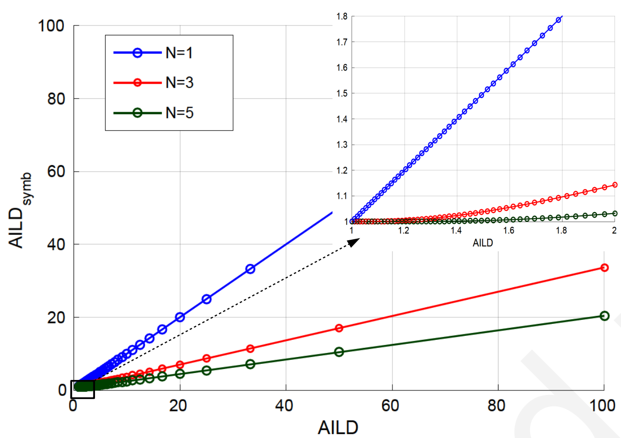


Fig. 11: $AILD_{symb}$ versus $AILD$ for different values of N and M = 1.

when N = 1, the $AILD_{symb}$ is directly proportional to $AILD$, with the constant of proportionality equal to M. For $N > 1$, the increase of the $AILD$ and/or decrease of N yields an increase of the $AILD_{symb}$ which can be approximated as linear for large $AILD$ values. Typically, it is easily concluded that $AILD_{symb} \rightarrow \frac{M}{N} AILD$ when $AILD >> 1$.

From this performance analysis, we investigate in the sequel of the study how to gauge and configure the coding FDP framework. Indeed, the ABL_{symb} and $AILD_{symb}$ have a direct effect on the file delivery performance, depending on the coding strategies exploited within the FDP protocol. High ABL_{symb} values and small $AILD_{symb}$ values may degrade the effectiveness of coding strategies and may also worsen the final perceptual quality. In the following, we provide a detailed analysis on these aspects. In that perspective, the next section introduces the various coding strategies available at the FDP level.

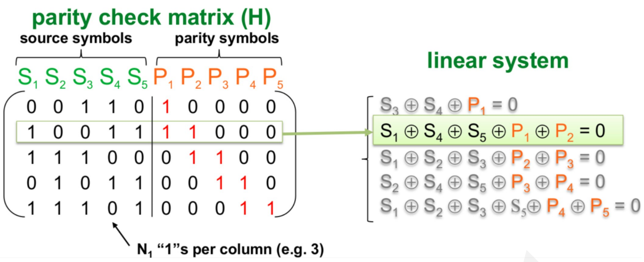


Fig. 12: Parity check matrix of LDPC-Staircase codes as defined in RFC5170 standard.

IV. Coding strategies within FDP for a file delivery

A. Description of coding strategies

For the file download service, it is important to deliver VoD files with a higher level of robustness even under the severe conditions in wireless channels. To address this issue, HbbTV adopts a framework of FDP with one of three coding strategies; employing only repetition code (denoted by Case-1), or only AL-FEC code (denoted by Case-2), or both repetition and AL-FEC mechanisms at the same time (denoted by Case-3).

Concerning AL-FEC code, the present document focuses on the so-called LDPC-Staircase code that belongs to the well-known class of binary structured LDPC codes and whose the construction of their parity check matrices[6] is very simple as described in RFC5170 standard [20]. As shown in Fig.12, in addition to length, size, and Pseudo-Random Number Generator (PRNG), this code is characterized by a parameter N1 which defines the degree of source variable nodes. As discussed and briefly analyzed in a previous study [12], this kind of AL-FEC code represents a good solution for the FDP protocol to protect the files of the Push-VoD service over HbbTV technology due to many interesting key features such as:

- It is a systematic code and guarantees very high encoding and decoding throughputs [21].
- It shows excellent performance for medium to large object sizes and also perform close to ideal codes in many use-cases when hybrid iterative/maximum likelihood (IT/ML) decoding[7] is performed at the receiver [20] [21].
- It is a good choice whenever the processing load at a software encoder or decoder must be kept to a minimum even using hybrid iterative/maximum likelihood (IT/ML) decoding [21].

[6]The parity check matrix is subdivided into two parts $(H_1|H_2)$, where H_1 is a sparse matrix with no particular structure, and H_2 is a staircase matrix (i.e., it is the part of redundant symbols) [19].

[7]Over an erasure channel, ML decoding is equivalent to resolving a linear system using the Gauss elimination (GE) method, which implies a cubic decoding complexity ($O(n^3)$). Hybrid decoding, proposed in [22], is an appropriate approach that allows a reduced complexity decoder with performance close to the optimal ML decoding. By this approach, we start by IT decoding, and in the case of decoding failure, we proceed with ML decoding (on a smaller-size or more-sparse parity check matrix)

- It is standardized as RFC5170 with FLUTE)/ALC protocol [23] to enable a reliable and scalable (no limit on the number of receivers) multicast/broadcast delivery of contents (any kind of files (e.g. multimedia)) over unidirectional transport networks (i.e. without feedbacks) [20]. It is also standardized as RFC6816 for robust real-time (streaming) delivery services (i.e., broadcasting continuous data stream) in [21]. Moreover, since mid 2012, this code is the AL-FEC scheme included in the ISDB-Tmm Japanese standard for mobile multimedia broadcasting where they are used along with FLUTE/ALC to improve the reliability and efficiency of push video services [10].

This FDP framework is characterized by different parameters which can influence the performance, such as

1) The number of symbols per FDP segment, M:
It is a key parameter of FDP. It intervenes on the shape of data sent and thus loss distribution of data which can has a direct impact on the performance of different recovery strategies within FDP. Therefore, we address the problem of the good dimensions of this parameter for the different strategies provided by FDP.
2) The File/FEC segments transmission mode, $Tx\text{-}mode\text{-}case$:
It is a key parameter of FDP. [5] precises that File (and FEC) segments may be sent in any order without giving a specific way for different recovery strategies to obtain a reliable FDP. Therefore, a suitable segment transmission mode for a given recovering strategy should be determined.
Moreover, the used LDPC-Staircase AL-FEC code creates strong relationships between repair symbols, each repair symbol is the sum of the previous repair symbol with some additional source symbols (see Fig.12). In that case the subset of received symbols does potentially impact the LDPC-Staircase AL-FEC code performance. Having a good symbol scheduling for LDPC-Staircase AL-FEC code is equivalent to finding a good File/FEC segments transmission mode within FDP.
3) The amount of FEC and/or repetition symbols:
They are key parameters of recovery strategies used within FDP. These two parameters have a significant influence on the FDP performance.

In the following section, an in-depth analysis of these parameters is provided to investigate how FDP protocol manages to recover from packet losses using LDPC-Staircase AL-FEC code and/or repetition scheme.

B. Definition of possible recovering areas of recovering strategies

Consider a file composed of k_{file} symbols, repetition code and/or AL-FEC code allows to generate n_{symb} symbols. By transmitting these symbols over broadcast network, the number of symbols actually received equals

$$n_{rcvd} = n_{symb}.(1 - SLR). \tag{31}$$

To success the file recovery, n_{rcvd} must be at least equal to k_{file} (i.e., $n_{rcvd} \geq k_{file}$). Based on the channel Gilbert model, we can determine the (p_{pckt}, q_{pckt}) areas supported by the coding strategies within FDP. Using Eq. (24), Eq. (8) and Eq. (31), the lowest value of q_{pckt} for possible recovering within FDP is given by

$$q_{pckt} \geq q_{pckt,min} = p_{pckt}.\frac{\frac{R}{(1-p_{pckt})^{N-1}}}{1 - \frac{R}{(1-p_{pckt})^{N-1}}},$$
$$\text{with } R = \begin{cases} R_{rep} = \frac{1}{N_{rep}} & : \text{for Case-1} \\ R_{FEC} = \frac{K_{FEC}}{N_{FEC}} & : \text{for Case-2} \\ R_{rep}.R_{FEC} & : \text{for Case-3} \end{cases} \tag{32}$$

where respectively K_{FEC}, N_{FEC}, and N_{rep} define the dimension, the length of the AL-FEC code and the number of times of sending k_{file}.

Fig. 13 provides the areas for possible recovering file for different values of R and N. Fig. 13a shows the areas for $N \in \{1, 2, 3, 5\}$ with R=$\frac{1}{3}$ and Fig. 13b shows the areas for $R \in \{1/5, 1/3, 1/2, 2/3\}$ with $N = 5$. From this figure we can conclude that the increase of R and N reduces the possible file recovering areas of coding strategies provided within FDP. These areas will be used in further analysis.

V. Performance analysis

A. Simulation environment

We evaluate the erasure recovery performance of the HbbTV based Push-VoD services using the FDP protocol. This performance has been assessed through Monte Carlo simulations (10^6 iterations) and are expressed in terms of decoding failure probability and reception overhead ε.

The decoding failure probability corresponds to the probability that erased file symbols cannot be all recovered given a set of received symbols. The reception overhead ε measures how many extra symbols over k_{file} are needed to recover the file by a receiver. This metric can be defined as follows:

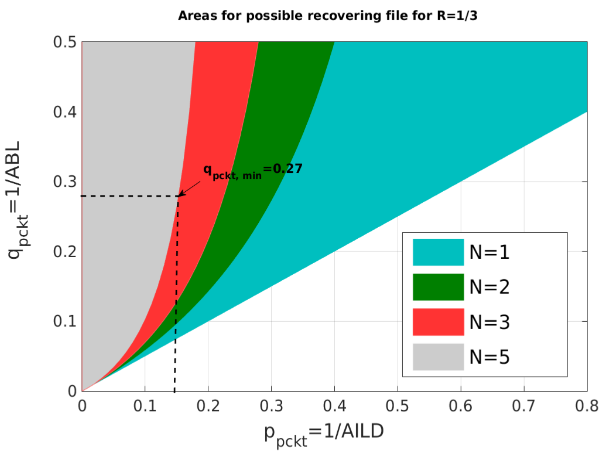
$$\varepsilon = \frac{E\{r\}}{k_{file}} - 1, \tag{33}$$

where $E\{r\}$ is the average amount of necessary symbols that allows to reconstruct the data file and,

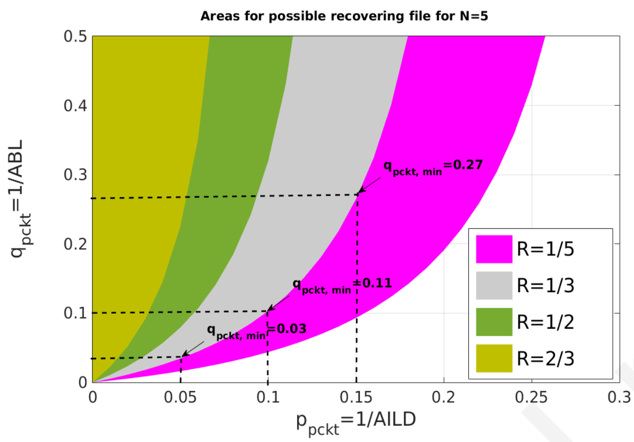
$$k_{file} \leq r \leq \begin{cases} \frac{k_{file}}{R_{rep}} & : \text{for Case-1} \\ \frac{k_{file}}{R_{FEC}} & : \text{for Case-2} \\ \frac{k_{file}}{R_{rep}.R_{FEC}} & : \text{for Case-3} \end{cases} \tag{34}$$

The file recovering will succeed from $k_{file}(1 + \varepsilon)$ received symbols where,

$$0 \leq \varepsilon \leq \varepsilon_{max} = \begin{cases} \frac{1-SLR}{R_{rep}} - 1 & : \text{for Case-1} \\ \frac{1-SLR}{R_{FEC}} - 1 & : \text{for Case-2} \\ \frac{1-SLR}{R_{rep}.R_{FEC}} - 1 & : \text{for Case-3} \end{cases} \tag{35}$$



(a)



(b)

Fig. 13: Possible file recovering areas within FDP. (a) for various values of N and $R = \frac{1}{3}$. (b) for various values of R and $N = 5$.

The lower ε, better is the recovery scheme. For perfect code, ε equals to 0. Therefore, $k_{file}.\varepsilon$ represents the number of symbols over k_{file} which are required to success the recovering.

To determine the best File (and FEC) segment scheduling for a reliable delivery of Push-VoD services, we introduce different possible transmission modes for each recovering strategy within FDP. Tables I, II, and III show these modes classified into different categories for Case-1, Case-2 and Case-3 respectively. In the sequel, we use $file\_sz$ = 9.2Kbytes and assume that $file\_seg\_sz$ and FEC_seg_sz are equal to 920 bytes. In addition, we use $M \in \{1, 5, 10, 20, 40, 184\}$. For that, Table IV describes number and length in bytes of symbols composed files for each value of M. In addition, based on numerical results showed in [24], we use $N1 = 5$ (cf. Section IV-A) to make a good compromise between the decoding complexity, the improvement of ML decoding, and the degradation of IT decoding.

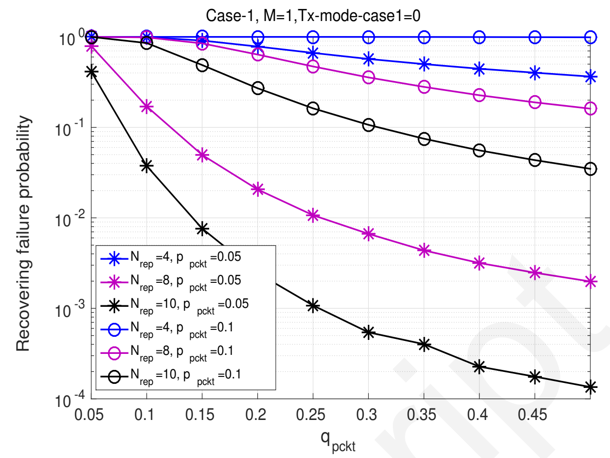


Fig. 14: Effect of N_{rep} parameter variation in Case-1 based FDP versus loss patterns (p_{pckt}, q_{pckt}) with $M = 1$.

All the results presented in the following sections were obtained by using $N = 5$. The resulting SLR is given in Fig. 6 and the areas for possible recovering file based on FDP for different values of rate R is shown in Fig. 13b. For instance, for R=$\frac{1}{3}$, the possible recovering area within FDP is started from $q_{pckt,min} = 0.27$ for p_{pckt}=0.15 and from $q_{pckt,min} = 0.11$ for p_{pckt}=0.1.

B. Investigation of coding strategies for a file delivery using FDP

The aim of this section is to find out the effect of the different parameters of the FDP framework (cf. Section IV) using the study shown in Section III. Preliminary simulation results given in [12] with LDPC staircase codes, and [13] with RS codes, based on a simple Gilbert channel model, have shown the sensitivity of the file delivery mechanisms to such parameters. Therefore, we investigate further the system performance, but exploiting the new expressions derived from the theoretical modeling of losses of FDP protocol presented in Section III. More precisely, we study the effect of changing the number of symbols per segment, M, and the effect of changing the File (and FEC) segments transmission modes, $Tx\text{-}mode\text{-}case$, for three coding strategies. In addition, we study how many repeated transmissions, N_{rep}, are needed to successfully receive the files for Case-1, and how impact the variation of R_{rep} and R_{FEC} on Case-3 performance.

1) Effect of parameter N_{rep}: This shows the number of repetitions N_{rep} required to rebuild a file using Case-1 based FDP for different packet loss patterns (p_{pckt}, q_{pckt}). M was fixed to 1 and the File segments are sent randomly.

Fig. 14 depicts the file recovering failure probability as a function of (p_{pckt},q_{pckt}) for $N_{rep} \in \{4, 8, 10\}$. This figure shows that the needed N_{rep} for low-loss environments is lower than the needed one for high-loss environments. For instance, for 10^{-3} as recovering failure probability,

TABLE I: The five File segment transmission modes ($Tx\text{-}mode\text{-}case1$) considered for Case-1.

Category (1)	**Fully random order**
Tx-mode-case1=0	all File segments are sent and repeated randomly
Category (2)	**File segments are sent sequentially and then repeated in a certain order**
Tx-mode-case1=1	repeated sequentially
Tx-mode-case1=3	repeated randomly
Category (3)	**File segments are sent randomly and then repeated in a certain order**
Tx-mode-case1=2	repeated sequentially
Tx-mode-case1=4	repeated randomly

TABLE II: The five File/FEC segment transmission modes ($Tx\text{-}mode\text{-}case2$) considered for Case-2.

Category (1)	**Fully random order**
Tx-mode-case2=0	all File/FEC segments are sent randomly
Category (2)	**File segment are sent first, sequentially, followed by FEC segment in a certain order.**
Tx-mode-case2=1	FEC segments are sent sequentially
Tx-mode-case2=2	FEC segments are sent randomly
Category (3)	**File segments are sent first randomly, followed by FEC segments in a certain order**
Tx-mode-case2=3	FEC segments are sent sequentially
Tx-mode-case2=4	FEC segments are sent randomly

TABLE III: The seven File/FEC segment transmission modes ($Tx\text{-}mode\text{-}case3$) considered for Case-3.

Category (1)	**Fully random order**
Tx-mode-case3=0	all File/FEC segments are sent and repeated randomly
Category (2)	**All File/FEC segments are sent sequentially and all then repeated in a certain order.**
Tx-mode-case3=1	repeated sequentially
Tx-mode-case3=3	repeated randomly
Category (3)	**All File/FEC segments are sent randomly and all then repeated in a certain order**
Tx-mode-case3=2	repeated sequentially
Tx-mode-case3=4	repeated randomly
Category (4)	**File segments are sent first sequentially, followed by FEC segments randomly and all then repeated in a certain order**
Tx-mode-case3=5	repeated sequentially
Tx-mode-case3=6	repeated randomly

TABLE IV: The $symb\_sz$ and the number of symbols composed files versus M parameter for $file\_sz$ = 9.2Kbytes and $file\_seg\_sz$=FEC_seg_sz = 920 bytes.

M	**1**	**5**	**10**	**20**	**40**	**184**
symb_sz(bytes)	920	184	92	46	23	5
Number of symbols	100	500	1000	2000	4000	184000

N_{rep}=8 is sufficient for environment with ($p_{pckt} = 0.05$, $q_{pckt} = 0.5$) whereas it is not sufficient for that with ($p_{pckt} = 0.05$, $q_{pckt} = 0.1$). In addition, from the figure we can notice that performance gets quite poor already with low PLR, ABL and SLR. For example with $PLR = 11.11\%$ and $ABL = 2.5$ ($p_{pckt} = 0.05$, $q_{pckt} = 0.4$, $SLR = 27.6\%$, $ABL_{symb} = 1.6851$) it can take 10 repetitions to receive the file with 10^{-4} as recovering failure probability.

The increase of N_{rep} improves performance at the cost of increasing total transmission time and $E\{r\}$.

2) Effect of parameter M: For this study, all segments are sent randomly (i.e., $Tx\text{-}mode\text{-}case1$=0 for Case-1, $Tx\text{-}mode\text{-}case2$=0 for Case-2, $Tx\text{-}mode\text{-}case3$=0 for Case-3). The use of this transmission mode is chosen so as to mitigate the burst losses which may occur at File (and FEC) segment level and enables to investigate correctly the effect of M.

The file recovering failure probability for Case-1 based FDP obtained as a function of (p_{pckt},q_{pckt}) for different M values with N_{rep}=10, is shown in Fig. 15. From this figure we can conclude that a modification of M does not impact the file recovering performance, whatever the packet loss patterns (p_{pckt}, q_{pckt}). Indeed, additional results not shown in this paper exhibit the same effect on $E\{r\}$.

As regards Case-2 based FDP, TABLE V and Fig. 16 provide respectively, the decoding failure probability and the reception overhead ε versus packet loss patterns (p_{pckt}, q_{pckt}) for

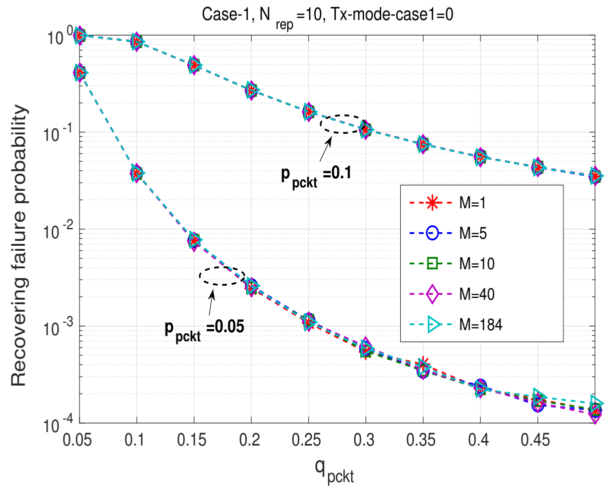
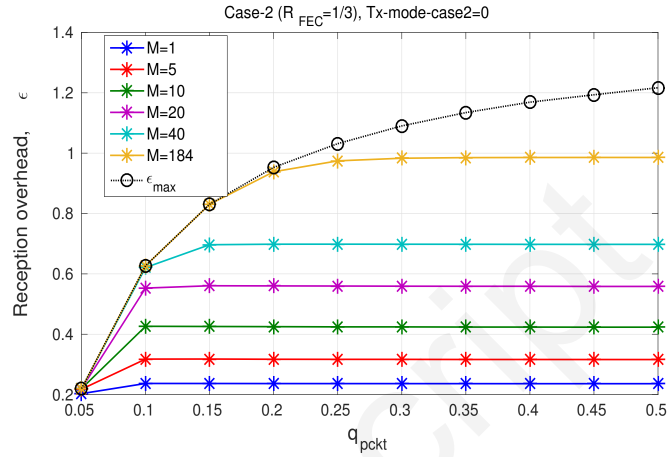


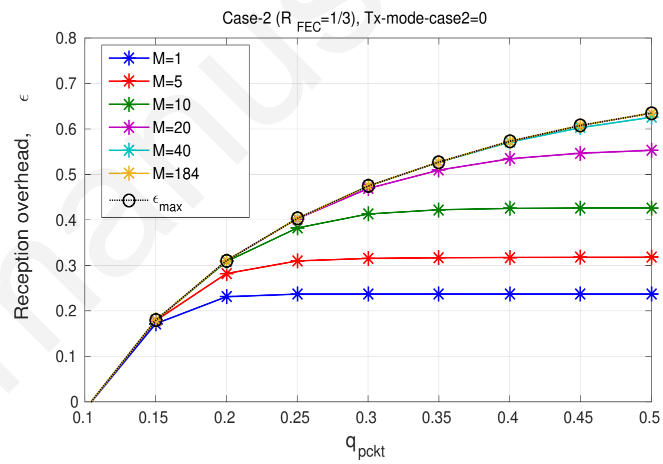
Fig. 15: Effect of parameter M on the recovering failure probability in Case-1 based FDP versus loss patterns (p_{pckt}, q_{pckt}) with $N_{rep} = 10$.

different M values with R_{FEC}=$\frac{1}{3}$.
This table shows that the increase of M increases the decoding failure probability for all (p_{pckt}, q_{pckt}) values. This is due to the fact that the loss of a segment leads to the loss of M consecutive symbols which causes burst-loss at symbol level and then stucks the LDPC-Staircase decoding. Therefore the increase of M increases the ABL_{symb} (cf. Section III-B2). In addition, an increase of M is equivalent to a decrease $symb\_sz$. It is most efficient to keep the symbol size as large as possible (i.e., M is low) so that on the one side, the file won't be protected by a short LDPC-Staircase code which may impact negatively on correction capabilities, and on the other side the total amount of symbols will be low and there will be also fewer lost symbols, which are tried to be recovered in the decoding. For instance, with $M = 1$ the file consists of 100 symbols of size 920 bytes and protected by a $(100, 300)$ LDPC-Staircase code whereas with $M = 40$ the file consists of 4000 symbols of length 23 bytes and protected by a $(4000, 12000)$ LDPC-Staircase code.
Fig. 16 shows that for low-loss environments the increase of M increases ε for each erasure channel case (p_{pckt}, q_{pckt}). This is explained by the fact that the decoder needs more data because it is faced with problem of decoding failure as shown in TABLE V. For high-loss environments, M parameter has no impact because the decoder can not succeed the decoding in all cases.

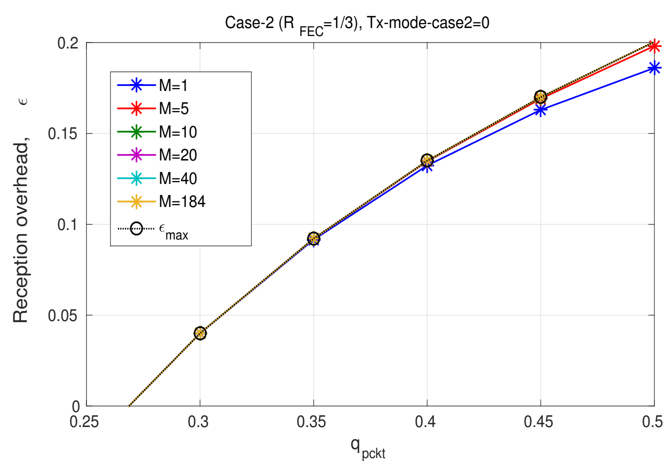
For Case-3 based FDP, Table VI and Fig.17 illustrate respectively, the result of the decoding failure probability and the reception overhead, ε, as a function of (p_{pckt}, q_{pckt}) of LDPC-Staircase code (R_{FEC}=$\frac{2}{3}$) combined with repetition (R_{rep}=$\frac{1}{2}$) for different M values.
Both show that the increase of M does not necessarily lead to increase the decoding failure probability, as seen with Case-2, thanks to the repetition technique. By increasing M



(a) p_{pckt}=0.05



(b) p_{pckt}=0.1



(c) p_{pckt}=0.15

Fig. 16: Effect of parameter M on the reception overhead in Case-2 based FDP versus loss patterns (p_{pckt}, q_{pckt}) with R_{FEC}=$\frac{1}{3}$. (a) for $p_{pckt} = 0.05$, (b) for p_{pckt}=0.1, and (c) for $p_{pckt} = 0.15$.

TABLE V: Effect of parameter M on the decoding failure probability in Case-2 based FDP versus loss patterns (p_{pckt}, q_{pckt}) with R_{FEC}=$\frac{1}{3}$.

(p_{pckt},q_{pckt}) \ M		1	5	10	20	40	184
p_{pckt}=0.05	q_{pckt}=0.05	1.74e-3	4.42e-1	9.74e-1	9.99e-1	1	1
	q_{pckt}=0.1	1.3e-5	3.58e-4	6.05e-4	2.08e-2	4.52e-1	1
	q_{pckt}=0.2	0	5e-6	1.2e-5	2.95e-4	6.23e-4	3.12e-1
	q_{pckt}=0.35	0	2.66e-6	5e-6	1.2e-5	7e-5	4.33e-3
	q_{pckt}=0.5	0	0	0	9.99e-6	1.5e-5	1.8e-3
p_{pckt}=0.1	q_{pckt}=0.05	1	1	1	1	1	1
	q_{pckt}=0.1	8.7e-1	9.99e-1	9.99e-1	1	1	1
	q_{pckt}=0.2	4.27e-4	8.41e-2	6.39e-1	9.91e-1	9.99e-1	1
	q_{pckt}=0.35	4.7e-5	9.97e-4	7.74e-3	2.53e-1	8.95e-1	1
	q_{pckt}=0.5	9e-6	2.96e-4	6e-4	2.23e-2	4.17e-1	9.99e-1
p_{pckt}=0.15	q_{pckt}=0.05	1	1	1	1	1	1
	q_{pckt}=0.1	1	1	1	1	1	1
	q_{pckt}=0.2	9.96e-1	9.99e-1	1	1	1	1
	q_{pckt}=0.35	2.09e-1	9.59e-1	9.99e-1	1	1	1
	q_{pckt}=0.5	8.43e-3	5.66e-1	9.7e-1	9.99e-1	1	1

from 1 to 10, the decoding failure probability has improved and thus the decoder needs fewer symbols. Beyond M=10, the decoding failure probability increases where the decoder needs more symbols to succed decoding.

3) Effect of the Tx-mode-case parameter: In this section, we investigate the impact of File (and FEC) segment transmission modes on the performance of Case-1, Case-2 and Case-3 within FDP. These modes are given in TABLE I, TABLE II and TABLE III for Case-1, Case-2 and Case-3 respectively. M is fixed to 1 to avoid the effect of burst-loss at symbol level induced by $M \neq 1$ (see Section. V-B2) and to focus only on the effect of the $Tx\text{-}mode\text{-}case$ parameter.
Fig. 18 provides recovering/decoding failure probability of Case-1 (with N_{rep}=10), Case-2 (with R_{FEC}=$\frac{1}{3}$) and Case-3 (with $R_{FEC} = \frac{2}{3}$) and $R_{rep} = \frac{1}{2}$) within FDP as a function of packet loss patterns (p_{pckt}, q_{pckt}) for different File (and FEC) transmission modes.
Fig. 18a shows that $Tx\text{-}mode\text{-}case1$ has no impact on the file recovery whatever the packet loss pattern (p_{pckt}, q_{pckt}). From Fig. 18b, we see that transmitting the FEC segments sequentially just after the File segments (sequentially ($Tx\text{-}mode\text{-}case2 = 1$) or randomly ($Tx\text{-}mode\text{-}case2 = 3$)), leads to very poor results for all packet loss patterns. The reason for this is that with the LDPC-Staircase AL-FEC code, the FEC symbols are produced by following the stairs of parity check matrix of the code (i.e., from the second FEC symbol, each FEC symbol depends on the previous one). In addition, for all packet loss patterns, it appears that $Tx\text{-}mode\text{-}case2$=0, $Tx\text{-}mode\text{-}case2$=2, and $Tx\text{-}mode\text{-}case2$=4 are the best and allow to disappear as possible the burst-losses which can be occured at segment level.
Fig. 18c shows that, for high loss environment (when p_{pckt} is greater than or equal to 0.10), $Tx\text{-}mode\text{-}case3$ modes behave exactly the same for all values of q_{pckt}. Whereas, there is a slight difference between $Tx\text{-}mode\text{-}case3$ modes when p_{pckt} is low (equals to 0.05) and whatever the value of q_{pckt} with the benefit to $Tx\text{-}mode\text{-}case3$=5. Then $Tx\text{-}mode\text{-}case3$=5 represents the best mode which allows to avoid the problem of burst-loss that can be generated at the File/FEC segment level.

Moreover, Fig. 19 and Fig. 20 illustrate the reception overhead, ε, of Case-2 and Case-3 respectively versus packet loss patterns (p_{pckt}, q_{pckt}) for the different associated transmission modes.
Fig. 19 shows that $Tx\text{-}mode\text{-}case2$=2 and $Tx\text{-}mode\text{-}case2$=4 need the lowest reception overhead in all cases. In addition, $Tx\text{-}mode\text{-}case2$=2 has an additional benefit over $Tx\text{-}mode\text{-}case2 = 4$ since File segments are sent in sequence, which avoids the extra delay required for the application to submit them all. Fig. 20 shows that $Tx\text{-}mode\text{-}case3$ has an impact on reception overhead only at low-loss environment (cf. Fig.20a and Fig.20b). At low-loss environment, it's obvious that $Tx\text{-}mode\text{-}case3$=0 gives the highest reception overhead since File/FEC segment are sent randomly. For the other $Tx\text{-}mode\text{-}case3$ modes, Fig.20a and Fig.20b show a little difference between the needed reception overhead with the benefits to $Tx\text{-}mode\text{-}case3$=1 and $Tx\text{-}mode\text{-}case3$=5 since File segments are sent first sequentially.

4) Effect of parameters R_{rep} and R_{FEC}: The following investigations are with $M = 1$ and $Tx\text{-}mode\text{-}case3$ = 0. This choice of parameters allows to mitigate the burst-loss that can occur at symbol level owing to respectively the loss of a segment composed of M symbols and the loss propagation from MPEG-4 TS packet level to FDP message level.
Fig. 21 provides the decoding failure probability in Case-3 of LDPC-Staircase code coupled with repetition as a function of packet loss patterns (p_{pckt}, q_{pckt}) for $R_{FEC} \in \{1/2, 2/3\}$ and $R_{rep} \in \{1/2, 1/4, 1/6\}$. This figure shows that adding an amount of FEC symbols induces better performance than adding a number of repetitions.

C. Synthesis

When transmitting a file of HbbTV Push-VoD services over a broadcast network, each individual receiver needs to

TABLE VI: Effect of parameter M on the decoding failure probability in Case-3 based FDP versus loss patterns (p_{pckt} ,q_{pckt}) parameters with R_{FEC}=$\frac{2}{3}$, R_{rep}=$\frac{1}{2}$.

(p_{pckt},q_{pckt}) \ M		**1**	**5**	**10**	**20**	**40**	**184**
p_{pckt}=0.05	q_{pckt}=0.05	8.56e-1	8.41e-1	8.39e-1	8.44e-1	8.69e-1	9.69e-1
	q_{pckt}=0.1	1.21e-3	2.74e-4	8.19e-4	2.65e-3	6.93e-3	1e-1
	q_{pckt}=0.2	3.5e-5	1e-5	9.99e-6	8.6e-5	1.62e-4	1.47e-3
	q_{pckt}=0.35	3.2e-6	1.96e-6	0	1.76e-5	3.99e-5	9.2e-5
	q_{pckt}=0.5	1e-6	0	0	1.59e-5	3.99e-5	9.2e-5
p_{pckt}=0.1	q_{pckt}=0.05	1	1	1	1	1	1
	q_{pckt}=0.1	9.99e-1	9.99e-1	1	9.99e-1	9.99e-1	1
	q_{pckt}=0.2	4.24e-1	3.66e-1	3.65e-1	3.85e-1	4.63e-1	8.26e-1
	q_{pckt}=0.35	7.76e-3	1.81e-3	3.9e-3	9.42e-3	2.88e-2	2.73e-1
	q_{pckt}=0.5	1.19e-3	2.76e-4	7.77e-4	2.60e-3	6.9e-3	1
p_{pckt}=0.15	q_{pckt}=0.05	1	1	1	1	1	1
	q_{pckt}=0.1	1	1	1	1	1	1
	q_{pckt}=0.2	1	1	1	1	1	1
	q_{pckt}=0.35	9.95e-1	9.95e-1	9.95e-1	9.95e-1	9.95e-1	1
	q_{pckt}=0.5	8.8e-1	8.7e-1	8.65e-1	8.69e-1	8.89e-1	1

receive all fragments of the file with a low total transmission time and a reasonable error rate so that it can complete the recovery via broadband channel using the File segment recovery technique. Therefore, the required total amount of data and the decoding failure probability represent two key metrics to determine the best coding strategies used within FDP.

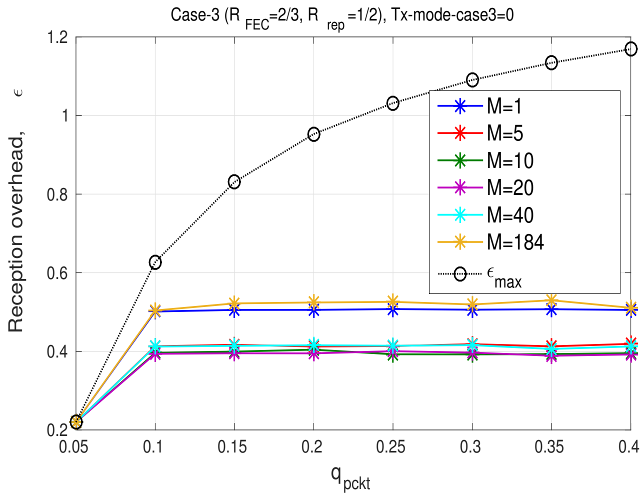
From the former results, Case-1 based FDP represents the worst case for the same amount of data transmitted in the system. More precisely, according to Fig. 14, Table II and Table III, Case-1 needs a high number of repeat transmission loops to obtain performance equivalent to those of Case-2 or Case-3. In addition, the receivers will likely receive more and more duplicate data as they attempt to obtain their last fragment of the file. In contrast, with Case-2 and Case-3, the file reception progress at the arrival rate of symbols and no time is wasted waiting to receive a specific symbol. Hence, the total amount of data and the time required to deliver files correctly is reduced compared to Case-1.
Consequently, Case-1 based FDP causes a transmission with a high amount of data to be sent, and thereby a download time that drastically increases with the symbol loss rate and the file size.

When File/FEC segments are sent randomly, Case-2 based FDP outperforms Case-3 based FDP in terms of decoding failure probability and reception overhead, but only when M is below a certain threshold (in our case below M=10). When M is high, the repetition technique helps Case-3 to reduce the decoding failure probability compared to Case-2. From Fig.16 and Fig.17, we observe that Case-2 needs more symbols and so more download time than Case-3 to succeed in the decoding step when M is greater than 10.
By using the best File/FEC segments transmission mode in each case, we can conclude that Case-2 achieves better decoding failure probability and requires a lower reception overhead than that of Case-3 whatever the loss-environment (see. Fig. 18b vs Fig.18c and Fig. 19 vs Fig. 20).
Regarding the recovery complexity, Case-3 has lower complexity than Case-2 since the former needs fewer operations and the repetition does not yield any additionnal complexity. To exceed Case-2, Case-3 needs more repeat transmission loops leading to high data rate and file recovery time.

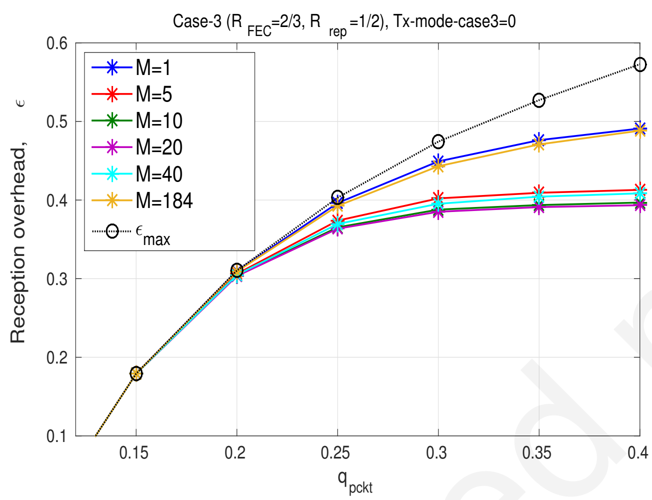
VI. CONCLUSION

In this paper, we focused on the HbbTV based Push-VoD services transmitted over DVB networks using the FDP protocol. This protocol is characterized by three levels of data representation and includes different coding strategies to ensure a reliable delivery. Due to these levels, the data shape and size change within this protocol give rise to three loss distributions that may affect the coding strategies behaviors. This article presents a reference work on the full feature investigation of FDP and how it handles a reliable file delivery on a wide variety of burst-erasure channels. Contrary to the massive majority of literature studies on the analysis of error correction mechanisms within transport protocols, our contribution is largely methodological and includes four main contributions.

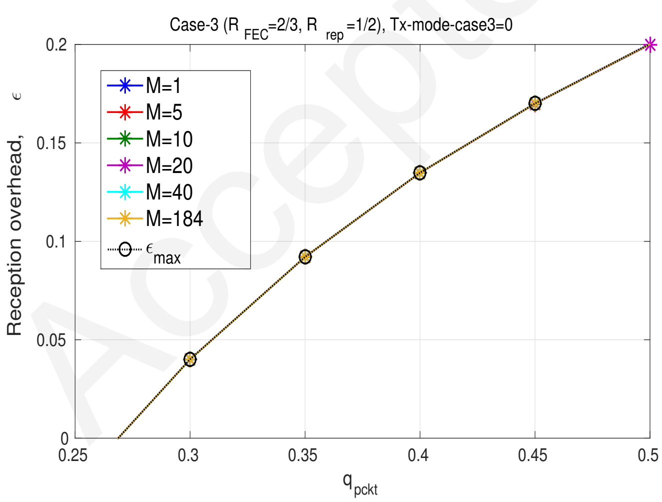
- First of all, we presented a thorough analytical analysis, based on a Markov chain, of the inter-layers loss spread within FDP on a wide variety of burst-erasure channels. This analysis enabled us to determine metrics allowing to measure the loss-run that helps to understand and tune effectively within FDP the different recovery strategies.
- Then, this study allowed to determine the possible file recovering areas for a coding strategy within FDP. We proposed and analyzed a first AL-FEC solution for a reliable Push-VoD service delivery via FDP over burst-loss channels. This solution is based on LDPC-Staircase codes due to its many interesting key features.
- Thereafter, an in-depth analysis of the configuration of coding FDP framework for a reliable Push-VoD service delivery over burst-loss channels was provided. We investigated the effect of different parameters in each coding strategy within FDP taking into account



(a) p_{pckt}=0.05

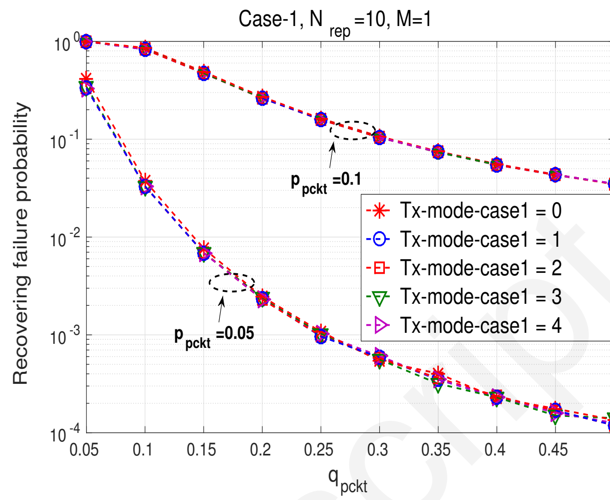


(b) p_{pckt}=0.1

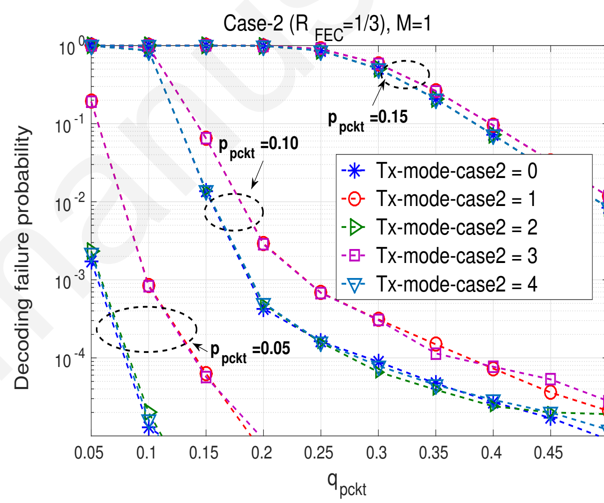


(c) p_{pckt}=0.15

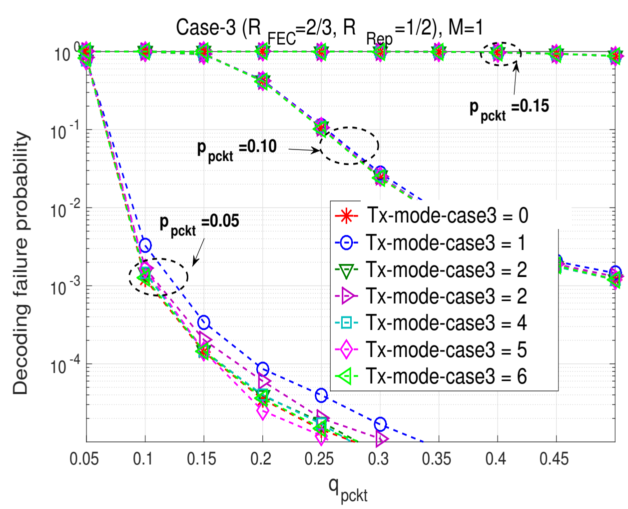
Fig. 17: Effect of parameter M on the reception overhead in Case-3 based FDP versus loss patterns (p_{pckt}, q_{pckt}) with R_{FEC}=$\frac{2}{3}$ and R_{rep}=$\frac{1}{2}$.



(a) Case-1



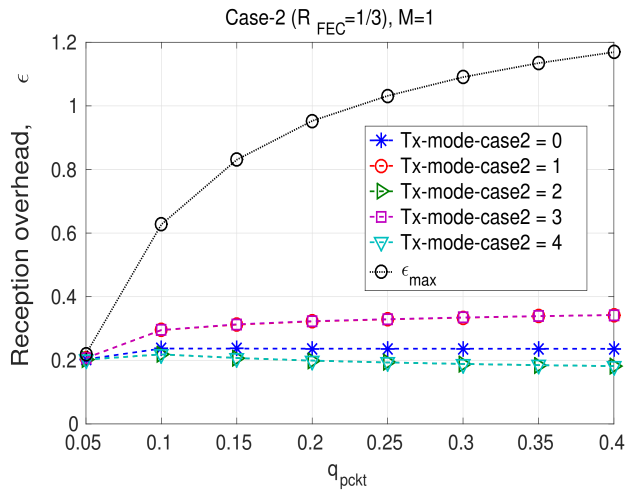
(b) Case-2



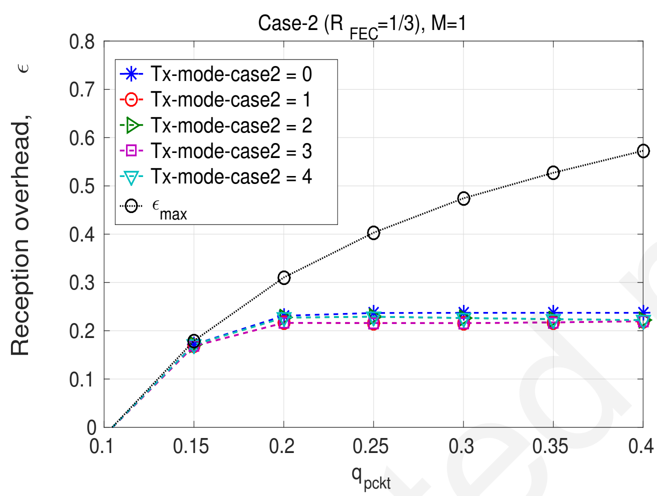
(c) Case-3

Fig. 18: Effect of the Tx-$mode$-$case$ parameter variation on the recovering/decoding failure probability versus packet loss patterns (p_{pckt}, q_{pckt})

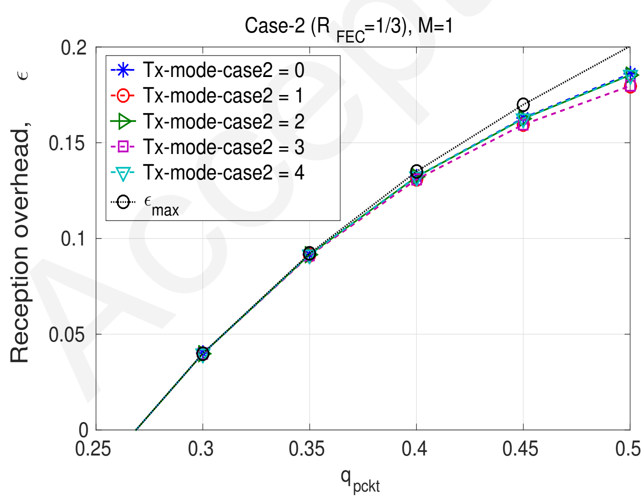
.



(a) $p_{pckt} = 0.05$

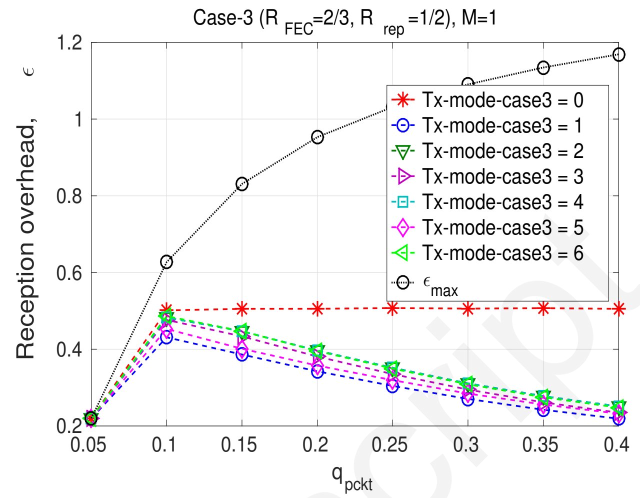


(b) $p_{pckt} = 0.1$

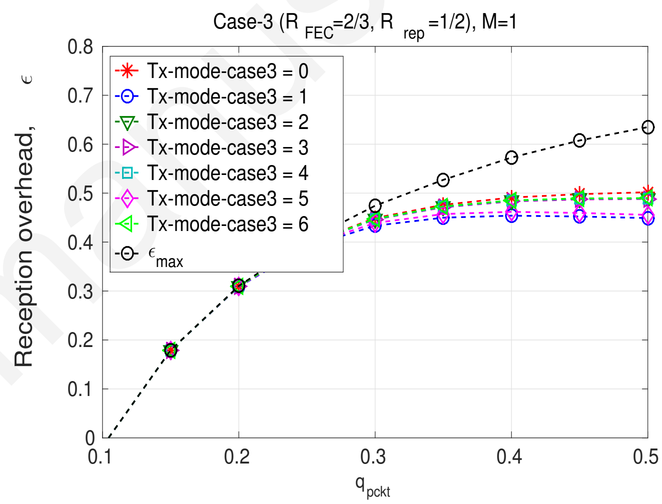


(c) $p_{pckt} = 0.15$

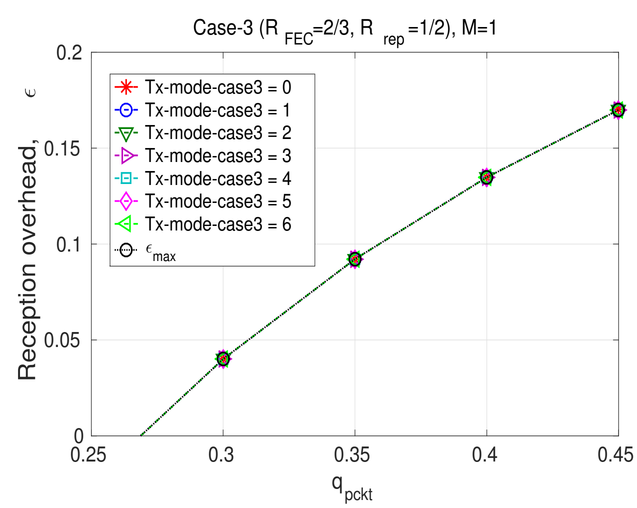
Fig. 19: $Tx\text{-}mode\text{-}case2$ parameter effect on reception overhead under different loss patterns (p_{pckt}, q_{pckt}) in Case-2.



(a) $p_{pckt} = 0.05$



(b) $p_{pckt} = 0.1$



(c) $p_{pckt} = 0.15$

Fig. 20: $Tx\text{-}mode\text{-}case3$ parameter effect on reception overhead under different loss patterns (p_{pckt}, q_{pckt}) in Case-3.

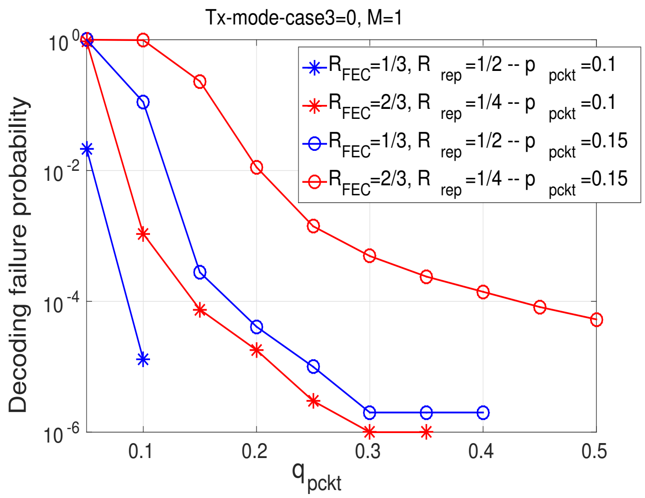


Fig. 21: Effect of R_{rep} and R_{FEC} parameters on decoding failure probability under different loss patterns (p_{pckt}, q_{pckt}) in Case-3.

correction capability and amount of data required to be sent to correctly receive a file. This study showed that FDP's parameters shall carefully be chosen to ensure a reliable broadcast of Push-VoD service.

- Finally, the suitable coding strategy within FDP that guarantees transmission robustness in front of correlated erasure channel impairments was presented. This study reveals that LDPC-Staircase AL-FEC code achieves the best trade-off between correction complexity, file recovering and download time compared to other recovery strategies provided by FDP.

Eventually, this paper provides an efficient model to assess the performance of the HbbTV system for non-linear service delivery using FDP. A detailed analysis was proposed to highlight the impact of the main parameters of the system with the example of an LDPC staircase AL-FEC. The developped analytical tools can be further exploited for the optimization of any other AL-FEC strategy, combined or not with repetion and segmentation processes.

Appendix A
Burst-loss modeling at FDP message level

Let us consider two consecutive FDP messages with states S_1 and S_2. Since each FDP message consists of N MPEG-4 TS packets, so S_1=$\{s_{1,k}, k = \overline{0, N-1}$[8]$\}$ and S_2=$\{s_{2,k}, k = \overline{0, N-1}\}$. $s_{1,k}$ and $s_{2,k}$ represent the state of the kth MPEG-4 TS packet within an FDP message with state S_1 and S_2 respectively. Therefore, the transition probability from S_1 to S_2 can be calculated as:

$$P(S_2/S_1) = prob(s_{2,0}/s_{1,N-1}).\prod_{k=1}^{N-1} prob(s_{2,k}/s_{2,k-1}), \quad (36)$$

[8] $\overline{0, N-1}$ represents the set of integers from 0 to N-1.

and the stationary probability of an FDP message with state S_1 is computed as:

$$\pi_{FDP}(S_1) = P(S_1) = prob(s_{1,0}).\prod_{k=1}^{N-1} prob(s_{1,k}/s_{1,k-1}), \quad (37)$$

where $prob(s_{1,0})$ and $prob(s_{i,k}/s_{j,k-1})$ are determined from $\mathbf{\Pi_{pckt}}$ and $\mathbf{P_{pckt}}$ respectively. The $\pi_{FDP}(S_2)$ is calculated in the same manner.

An FDP message has a no-loss state iff all the associated N MPEG-4 TS packets have no-loss states. This is defined by a single case denoted by T. An FDP message has a loss state iff at least one of the associated N MPEG-4 TS packets has loss state. There are $2^N - 1$ possible FDP loss states, depending of the MPEG-4 TS packet losses positions within the FDP message. These cases are denoted by F_i, $i \in \{0, \ldots, 2^N - 2\}$. Therefore, we can model the FDP messages distribution by a Markov chain with 2^N states characterized by the stationary probabilities vector

$$\mathbf{\Pi_{FDP}} = [\pi_{FDP}(T), \pi_{FDP}(F_0), \ldots, \pi_{FDP}(F_{2^N-2})], \quad (38)$$

where $\pi_{FDP}(T)$ and $\pi_{FDP}(F_i)$ are calculated using Eq. (37), and a transition probabilities matrix, $\mathbf{P_{FDP}}$ composed of $(2^N)^2$ transition probabilities computed using Eq. (36). This model can be reduced to a Markov chain with two states (T and F= $\bigcup_{i=0}^{2^N-2} F_i$) and characterized by the transition probabilities matrix

$$\mathbf{P^r_{FDP}} = \begin{bmatrix} P_{FDP}(T/T) & P_{FDP}(F/T) \\ P_{FDP}(T/F) & P_{FDP}(F/F) \end{bmatrix} = \begin{bmatrix} 1-p_{FDP} & p_{FDP} \\ q_{FDP} & 1-q_{FDP} \end{bmatrix}, \quad (39)$$

and the stationary probability vector

$$\mathbf{\Pi^r_{FDP}} = [\pi_{FDP}(T), \pi_{FDP}(F)], \quad (40)$$

where $\mathbf{\Pi^r_{FDP}}.\mathbf{P^r_{FDP}}$=$\mathbf{\Pi^r_{FDP}}$ and $\pi_{FDP}(T)$+$\pi_{FDP}(F)$=1.

The stationary probability of an FDP message with no-loss state (T) is computed using Eq. (6) as follows:

$$\pi_{FDP}(T) = \pi_{pckt}(T)(P_{pckt}(T/T))^{N-1} = \frac{q_{pckt}(1-p_{pckt})^{N-1}}{p_{pckt}+q_{pckt}} \quad (41)$$

Therefore, the stationary probability of an FDP message with loss state (F) is given by:

$$\pi_{FDP}(F) = \sum_{i=0}^{2^N-2} \pi_{FDP}(F_i) = 1 - \pi_{FDP}(T) = \frac{p_{pckt} + q_{pckt}[1-(1-p_{pckt})^{N-1}]}{p_{pckt}+q_{pckt}} \quad (42)$$

The transition probability $P_{FDP}(T/T)$ is calculated using Eq. (36) as follows:

$$P_{FDP}(T/T) = (P_{pckt}(T/T))^N = (1-p_{pckt})^N = 1 - p_{FDP}. \quad (43)$$

Consequently, the transition probability $P_{FDP}(F/T)$ is given by:

$$P_{FDP}(F/T) = 1 - P_{FDP}(T/T) = 1 - (1 - p_{pckt})^N \\ = p_{FDP}. \quad (44)$$

The transition probability $P_{FDP}(T/F)$ is computed by:

$$P_{FDP}(T/F) = \frac{prob(T \cap F)}{\pi_{FDP}(F)} = \frac{\sum_{i=0}^{2^N-2} prob(T \cap F_i)}{\pi_{FDP}(F)} \\ = \frac{\sum_{i=0}^{2^N-2} \pi_{FDP}(F_i).P(T/F_i)}{\pi_{FDP}(F)} \quad (45)$$

Whereas, by ordering F_i so that F_i ends with a packet with F state when $i = \overline{0, 2^{N-1} - 1}$ and with a packet with T state when $i = \overline{2^{N-1}, 2^N - 2}$ we can obtain that:

$$P(T/F_i) = \begin{cases} q_{pckt}(1 - p_{pckt})^{N-1}, & \forall i = \overline{0, 2^{N-1} - 1} \\ (1 - p_{pckt})^N, & \forall i = \overline{2^{N-1}, 2^N - 2} \end{cases} \quad (46)$$

and

$$\begin{cases} \sum_{i=0}^{2^{N-1}-1} \pi_{FDP}(F_i) = \pi_{pckt}(F) = \frac{p_{pckt}}{p_{pckt}+q_{pckt}} \\ \sum_{i=2^{N-1}}^{2^N-2} \pi_{FDP}(F_i) = \pi_{pckt}(T) - \pi_{FDP}(T) \end{cases} \quad (47)$$

Using Eq. (41), Eq. (46) and Eq. (47), Eq. (45) becomes:

$$P_{FDP}(T/F) = \frac{q_{pckt}(1 - p_{pckt})^{N-1}[1 - (1 - p_{pckt})^N]}{p_{pckt} + q_{pckt}[1 - (1 - p_{pckt})^{N-1}]} \\ = q_{FDP}. \quad (48)$$

Consequently, the transition probability $P_{FDP}(F/T)$ is given by:

$$P_{FDP}(F/F) = 1 - P_{FDP}(T/F) \\ = \frac{p_{pkt} + q_{pkt} - q_{pkt}(1 - p_{pkt})^{N-1}[2 - (1 - p_{pkt})^N]}{p_{pkt} + q_{pkt}[1 - (1 - p_{pkt})^{N-1}]} \\ = 1 - q_{FDP}. \quad (49)$$

REFERENCES

[1] ETSI TS 102 543 (V1.1.1), "Digital video broadcasting (DVB); globally executable MHP (GEM) specification 1.3 (including OTT and hybrid broadcast/broadband)," March 2008.

[2] ETSI TS 102 796 (V1.1.1), "Hybrid broadcast broadband TV," 2010.

[3] OIPF (V1.1.1), "Open IPTV forum release 1 specification," *Declarative Application Environment v1.1*, vol. 5, 2010, Available from http://www.oipf.tv/downloads.html.

[4] ETSI TS 102 809 (V1.1.1), "Digital video broadcasting (DVB); signalling and carriage of interactive applications and services in hybrid broadcast/broadband environments," 2010.

[5] ETSI TS 102 796 (V1.3.1), "Hybrid broadcast broadband TV," 2015.

[6] ETSI TR 102 993 (V1.1.1), "Digital video broadcasting (DVB); upper layer FEC for DVB systems," Feb. 2011.

[7] ETSI TS 102 472(V.1.3.1), "Digital video broadcasting (DVB); IP datacast over DVB-H: Content delivery protocols," 2009.

[8] ETSI TS 102 591-2, "Digital video broadcasting (DVB); IP datacast: Content delivery protocols (CDP) implementation guidelines; part 2: IP datacast over DVB-SH," 2010.

[9] ETSI TS 102 542-5, "Digital video broadcasting (DVB); guidelines for the implementation of DVB-IPTV phase 1 specifications; part 5: Content download system (CDS)," 2016.

[10] A. Yamada, H. Matsuoka, T. Ohya, R. Kitahara, J. Hagiwara, and T. Morizumi, "Overview of isdb-tmm services and technologies," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB11)*, pp. 274–282, Jun. 2011.

[11] F. Mattoussi, G. Zaharia, M. Crussière, and J.-F. Hélard, "Analytical modeling of losses in FDP protocol of HbbTV based Push-VoD services over DVB networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2017.

[12] F. Mattoussi, M. Crussière, and J.F. Hélard, "HbbTV based push-vod services over DVB networks: Analysis and AL-FEC code application," *IEEE International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*, Oct. 2016, Lisbon, Portugal.

[13] F. Mattoussi, M. Crussière, J.F. Hélard, and G. Zaharia, "A reliable delivery of HbbTV based Push-VoD services over DVB networks using RS codes," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2017, Cagliari, Italy.

[14] ETSI EN 301 192 (V1.6.1), "Digital video broadcasting (DVB); DVB specification for data broadcasting," 2015.

[15] ISO/IEC 13818-1, "Generic coding of moving pictures and associated audio information–part 1: Systems," 2013.

[16] E. N. Gilbert, "Capacity of burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1266, 1960.

[17] H. Sanneck, G. Carle, and R. Koodli, "A framework model for packet loss metrics based on loss runlength," *SPIE/ACM SIGMM Multimedia Computing Network Conference*, 2000.

[18] C. Jiao, L. Schiebert, and B. Xu, "On modeling the packet error statistics in bursty channels," *Proceeding of the* 27^{th} *IEEE conference on Local Computer Networks (LCN)*, pp. 534–541, 2002.

[19] D. Divsalar, H. Jin, and R. J. Mceliece, "Coding theorems for turbo-like codes," *IEEE Annual Allerton Conference on Communication, Control, and Computing*, pp. 201–210, Sept. 1998, Monticello, IL, USA.

[20] V. Roca, C. Neumann, and D. Furodet, "Low density parity check (LDPC) Staircase and triangle forward error correction (FEC) schemes," *IETF Request for Comments, RFC 5170 (Standards Track/Proposed Standard)*, 2008.

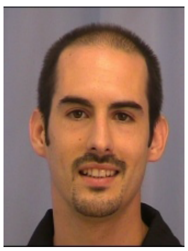
[21] V. Roca, M. Cunche, and J. Lacan, "Simple low-density parity check (LDPC) staircase forward error correction (FEC) scheme for FECFRAME," *IETF Request for Comments, RFC 6816 (Standards Track)*, Dec. 2012.

[22] E. Paolini, M. Varrella, M. Chiani, B. Matuz, and G. Liva, "Low-complexity LDPC codes with near-optimum performance over the BEC," *IEEE Conference on Advanced Satellite Mobile Systems (ASMS)*, pp. 274–282, Apr. 2008, Bologna, Italy.

[23] T. Paila, R. Walsh, M. Luby, V. Roca, and R. Lehtonen, "FLUTE-file delivery over unidirectional transport," *IETF Request for Comments, RFC 6726 (Standards Track) (obsoletes RFC 3926)*, Nov. 2012.

[24] M. Cunche and V. Roca, "Optimizing the error recovery capabilities of LDPC-Staircase codes featuring a Gaussian elimination decoding scheme," *IEEE International Workshop on Signal Processing for Space Communications (SPSC)*, pp. 1–7, Oct. 2008, Rhodes Island, Greece.

Ferdaouss Mattoussi received the Eng. degree in Telecommunications Engineering and the M.Sc. degree in Communication Systems from ENIT, Tunis, Tunisia, in 2008 and 2009 respectively. From 2010 to 2014, she was a PhD student at INRIA and Alcatel-Lucent Bell-Labs, France and received the Ph.D. degree in Computer Science for Digital Communications from the Joseph-Fourier University of Grenoble, France. From 2014 to 2015, she was a lecturer in signal processing at ECM, Marseille, France. In 2016, she joined the Digital Communications Systems team of the Telecommunications department of INSA, Rennes as an (R&D) Engineer in connected TV. Since Mai 2018, Ferdaouss Mattoussi works at Nokia Paris-Sacly as an (R&D) Engineer in 5G.



Matthieu Crussière received the M.Sc. and Ph.D. degrees in electrical engineering from the National Institute of Applied Sciences (INSA), Rennes, France, in 2002 and 2005, respectively. In 2005, he joined the Department of Telecommunications and Electronic Engineering at INSA as an Associate Professor and since this date, he has been with the Digital Communication Department of the Electronics and Telecommunications Institute of Rennes (IETR). In addition, in 2014, he started collaborations as an associate researcher at the Institute of Research and Technology (IRT) B-COM in Rennes. His main research interests lie in digital communications and signal processing techniques. His first works were focused on the optimization of high-bit rate powerline communications using hybrid multicarrier and spread-spectrum waveforms. Then he developed an expertise in adaptive resource allocation, optimization algorithms and system design for multicarrier and multiantenna systems. During the last years, he applied his research results to the optimization of broadcast, broadband and hybrid networks. He is author or co-author of more than 100 technical papers in international conferences and journals. He has been involved in several European and French national research projects in the field of powerline communications, broadcasting systems, ultra wideband and mobile radio communications.

Professor Hélard received his Dipl.-Ing. and his Ph.D in electronics and signal processing from the National Institute of Applied Sciences (INSA) in Rennes in 1981 and 1992 respectively. From 1982 to 1997, he was research engineer and then head of channel coding for the digital broadcasting research group at France Telecom Research Center (Orange Labs) in Rennes. In 1997, he joined INSA Rennes, which is one of the Grandes Ecoles in France, where he is today Full Professor, Classe Exceptionelle, which is the highest rank. He was Director of Research of INSA Rennes during 3 years from december 2010 to december 2013. He was also during 8 years Deputy Director of the Rennes Institute for Electronics and Telecommunications (IETR, UMR CNRS 6164), which is an academic research laboratory of 400 people, created in 2002 in association with the CNRS. His research interests lie in signal processing techniques for digital communications, such as space-time and channel coding, multi-carrier modulation, as well as multi-user communications and cross-layer techniques. He is involved in several European and national research projects in the fields of digital video terrestrial broadcasting, mobile radio communications and cellular networks, power-line and ultra-wide-band communications, cooperative communications and relaying techniques. Prof. J-F. Hélard is a senior member of IEEE, author and co-author of more than 270 technical papers in international scientific journals and conferences, and holds 15 European patents.

Gheorghe Zaharia received the B.S. degree in electronics and telecommunications from the Polytechnic Institute of Iasi, Romania, in 1981 and the D.E.A. and Ph.D. degrees in telecommunications, signal processing and electronics from the Institut National des Sciences Appliques (INSA) of Rennes, France, in 1991 and 1997, respectively. From 1983 to 1990 he was a Lecturer at the Technical University of Iasi, Romania. Between 1983 and 1985 he studied mathematics at the University Al. I. Cuza of Iasi. Since 1997, he is Associate Professor in the Department Communications Systems and Networks, the former Electronics and Communications Systems Department, INSA, Rennes. His teaching and research interests mainly concern the study of radio waves propagation, communications systems, information theory, and coding. Gheorghe Zaharia is Senior Member IEEE and member of the Romanian Society of Medical Bioengineering. He was member of the Scientific Committee of several international conferences (EHB 2011, 2013, 2015, 2017, MMS 2013, SENSET 2017,). He has more than 30 journal papers, about 120 papers presented at international conferences and 2 patents. He is also co-author of 2 books and 5 book chapters.