Project acronym: CONCERTO

Project full title: Content and cOntext aware delivery for iNteraCtive multimEdia healthcaRe applicaTiOns

Grant Agreement no.: 288502

Deliverable 4.1

Description and evaluation of context-awareness techniques

Contractual date of delivery to EC:	T0+9
Actual date of delivery to EC:	30/08/2012
Version:	1.0

Lead Beneficiary:	TCS
Participants:	TCS, VTT, CNIT, KU

Estimated person months:	N/A
Dissemination Level:	*PU*
Nature:	*R*
Total number of pages:	65

Keyword list:HTTP streaming, FEC, multi-camera system, camera selection, OFDMA, HARQ, prioritization, QoE, LDPC

Executive Summary

This Deliverable provides a description and an evaluation of context-awareness techniques. The main purpose of this deliverable is to give an overview of the considered technologies and detailed description of the corresponding elements for context-awareness techniques. In addition, this deliverable will also present some preliminary design and specification considerations related to context-awareness techniques such as fusion and protection studied and developed in the CONCERTO project. This document is structured as follows:

Chapter 2 is focused on HTTP streaming. Advantages and disadvantages of the various existing techniques of HTTP streaming in video delivery are analyzed. In addition the section also discloses the highlights of existing rate control algorithms in HTTP streaming. Finally it describes our research activities on HTTP Streaming. A new technique of cross-layer prioritization for HTTP Streaming is proposed. Taking into consideration the fact that retransmission generates delays and de-synchronization and decreases the video quality, it purposes to plan and to send at the Medium Access Control (MAC) layer directly the necessary number of packets to avoid retransmission.

In Chapter 3, our recent research activities on multi-camera system are presented. After an introduction and a State-of-the-art of Visual sensor networks, a preliminary camera selection technique for healthcare and safety applications in emergency areas is introduced. This technique has the objective to maximize the Quality of Experience(QoE) and to optimize the usage of radio and distributed resources.

In Chapter 4, our investigations in the context of packet-level coding and its benefits in healthcare, safety and emergency scenarios are detailed. Two directions are presented: the first one is relevant to packet-level codes constructed on Galois fields of order larger than 2 for erasure recovery, under efficient maximum likelihood (ML) decoding. This analysis of non-binary codes based on sparse parity-check matrices is fundamental to characterize the ultimate performance achievable with these codes and, as a consequence, to evaluate the possible advantages offered by this kind of solutions in the context of CONCERTO project. The second one concerns packet-level codes for the simultaneous correction of errors and erasures with an enhanced decoder for low density parity check (LDPC) codes and its capacity to correct bit-errors possibly present in the received stream which can improve the experienced multimedia communication quality .

In Chapter 5, 3D video delivery over wireless systems based on Orthogonal Frequency Division Multiple Access (Orthogonal Frequency Division Multiple Access (OFDMA)) is addressed by considering a MAC layer scheduling method combined with a prioritized queuing mechanism. The purpose is to prioritize the most important video components/layers with the goal of improving the perceived quality of 3D video at the receiver.

Finally, a chapter dedicated to conclusions completes this deliverable.

Contents

List of Figures

1 Introduction

The awareness of the context in which data should be transmitted gives the possibility to improve the effectiveness of the transmission. According to the context the media can be adapted and the data to send can be reshaped in order to improve the Quality of Experience (QoE) and Quality of Service (QoS) at the receiver. Several techniques have been studied in the literature to take advantage of the context awareness in different ways. It is, for example, possible to increase or decrease the protection of critical data according to the channel conditions or adapt the data rate following the expected available bandwidth.

The smart use of context awareness is one of the key objectives pursued in the construction of the CONCERTO content transmission architecture.

This deliverable covers a large panorama of context-awareness techniques, introducing the actual state of the art and presenting some preliminary studies and considerations that will be further developed in the WP4. The different sections of the deliverable analyze various techniques in different contexts of interest for the CONCERTO project, focalizing in particular on prioritization techniques (for HTTP streaming and for 3D video), correction codes, rate control algorithms, selection strategies for multi camera system and adaptive techniques for HTTP.

This deliverable will be the starting point of future activities of WP4.

2 HTTP Streaming

This section introduces HTTP streaming in video delivery, its' advantages and disadvantages. The section focuses especially on the possibilities of bitstream adaptation to various end terminals in heterogeneous networks where available bandwidth can vary several megabytes during the video streaming. This section also discloses the highlights of rate control algorithms in HTTP streaming.

2.1 Introduction

Download-based video transmission (pull technology) is nowadays probably the most common technique to distribute video in Internet. Some video services demand that the whole file is downloaded at the beginning to the target device before the actual playback, causing a possibility to long start-up delay. TCP is often used to control the flow since it is reliable protocol and guarantees the video delivery. However, this can cause long delays and impressive load on server especially when handling multiple requests.

Large video distribution platforms, such as Youtube, use so called progressive download, which enables to begin the video playback while still downloading the rest of the file. This is quite efficient method, when the target user has sufficient speed for the network connection. Naturally if the connection speed is low, annoying pauses in the playback will occur, which yields to very annoying quality of experience. The pull technology is difficult to be utilised in real-time video streaming (such as live feed from medical operation), e.g. because of long delays caused by TCP. Additionally, HD content demands high bandwidth and sufficient amount of data is needed into the buffer before the actual playback can start.

As the progressive download is still one of the most favourite streaming technologies, also HTTP streaming-based solutions have raised their heads in the past few years. The large overhead of TCP, caches, NATs and firewall issues in RTP/RTSP/UDP streaming are less problematic in HTTP [77]. The evolved Internet infrastructure support efficient usage of HTTP and Content Delivery Networks(Content Delivery Network (CDN)) [74], which can also utilise HTTP streaming in video delivery to clients. Furthermore, the additional advantages in HTTP streaming include matters such as

- No traditional streaming server is required
- Client can access and play the content instantly
- Simple setup
- Clients can seek to points which are not yet downloaded

However, HTTP streaming has also its disadvantages. The client compatibility is still quite low compared e.g. to Flash Media (commercial) [24]. Furthermore, the start-up latency can be quite long depending on the segment length (duration) and bitrate. In RTP/RTSP the latency is smaller than couple of seconds, which emphasises its usage in real-time streaming [24]. However, when delivering content via RTP/RTPS, there is no guarantee of the packet delivery which means that error handling routines are needed, such as error concealment in the video decoder or adaptation techniques [78].

The basic idea in all of the HTTP streaming solutions is similar; video file is splitted into segments and manifest file tells which segments are available for downloading. Most of the methods use Extensible Markup Language (XML) format for the manifest file, but HTTP Live Streaming (HLS) uses .m3u8 format. Two media formats are used for the content; MPEG-2(.ts) and **ISOMBFF!** (**ISOMBFF!**)(.mp4) The client downloads and plays the short segments, usually only few seconds long, of the original stream in the order defined in the playlist. The segments are encoded to multiple different data rates. The segmented streaming solution can be used for both live and on-demand, but the main advantage lies in the capability to adapt the prevailing network bandwidth e.g. by measuring the length of client's input buffer. For example, the client software can change the data rate based on the recent trends in measured network throughput. Fig. 1 presents an illustration of HTTP streaming scheme.

HTTP Streaming alongside with HTTP live streaming can be very useful alternatives for substituting progressive downloads. However, some work is left to do, since there are different platforms and operating systems that do not yet support HTTP streaming. The main advantages lie in adaptive streaming where the server knows the available bandwidth and clients device capabilities during the playback and therefore the bitrate adaptation is possible to achieve.

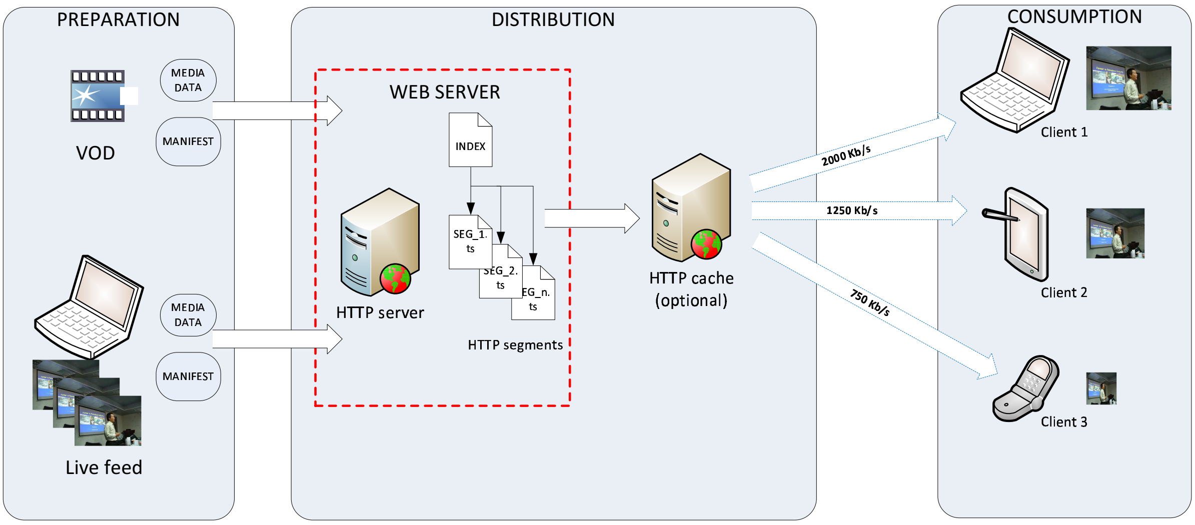


Figure 1: Overall principle of HTTP streaming.

There is no single standardized method for HTTP streaming. Different standardisation solutions from this area have been introduced for HTTP streaming, such as Apple HTTP Live Streaming (HLS) [4], Microsoft Smooth Streaming [6] and Dynamic Adaptive Streaming over HTTP(MPEG-DASH) [5]. The next subsection will introduce these HTTP streaming methods in detail.

2.2 Adaptive HTTP Streaming

The prevailing channel state is naturally one of the key elements when transmitting video content. High bitrate video stream in low bandwidth channel will cause packet droppings and therefore bad quality of experience for the end user. In traditional video streaming this means that either error resilience methods, error concealment or adaptation algorithms are needed. One solution is packet re-sending, but this forms delay problems especially in real-time streaming.

HTTP streaming enables adaptive streaming where the video content is encoded at multiple data rates and the best version according to available bandwidth can be streamed to client. Naturally, the channel state needs to calculated e.g. by monitoring the size of client's input buffer (small input buffer means narrow bw in the channel). After this, the playlist can be updated and client asks for smaller bitrate segment. Furthermore, if multiple clients with different terminal characteristics exists, client can ask for smaller resolution segments with smaller frame rate.

2.2.1 Smooth Streaming

Microsofts solution for adaptive HTTP streaming is called Smooth Streaming [6] and it has been introduced already in 2008 Summer Olympics. It was included in Microsoftś Internet Information Services (IIS) 7.0. The media format is a proprietary format as an extension of the ISO Base Media File Format. The manifest format is client manifested file, which is proprietary XML document and the solution has also server manifested file (SMIL document). Content is delivered using a Smooth Streaming-enabled IIS origin server.

Normal HTTP server is used as a web server for HLS and DASH, but Microsoft Smooth Streaming uses Microsofts own IIS as a web server. IIS enables adaptive streaming of media to Silverlight and other clients over HTTP by dynamically monitoring local bandwidth and video rendering performance. This yields to optimized content playback where the video quality is switched in real-time. For example, users with high bandwidth connections and modern computers can experience full HD 1080p quality streaming, while others with lower bandwidth or older computers can receive a lower quality stream in order to match better their bandwidth capability & device performance.

The concept of delivering video is similar as in other HTTP streaming utilising solutions. IIS delivers usually small fragments, typically couple of seconds long in the actual video, in a certain time window. If one video fragment does not meet the requirements, the next fragment will be delivered with lower quality containing smaller number of bytes and therefore requires also smaller bandwidth.

2.2.2 3GP-DASH

3GPP was the first standardisation body which introduced a solution for adaptive HTTP streaming (AHS) [1] based on stream segmentation, where the multimedia (audio, video) file is divided into small segments. The media format is an extension to ISO Base Media File Format (ISOBMFF) and the manifest format is Media Presentation Description (Media Presentation Description (MPD)). Similarly as in Microsoft Internet Information Services (IIS), MPD is the manifest file as a XML document that contains information where segments are located. The first version of the standard was released in 3GPP Release 9 on March 2010.

3GP-DASH is defined as a general framework independent of the data encapsulation format and it supports fast initial startup and seeking, adaptive bitrate switching, re-use of HTTP origin and cache servers, re-use of existing media playout engines, on-demand, live and time-shifted delivery.

2.2.3 MPEG-DASH

MPEG Dynamic Adaptive Streaming over HTTP (Dynamic Adaptive Streaming over HTTP (DASH)) [5] is a standard introduced by ISO/IEC MPEG group (ISO/IEC 23009-1) and it has gained more and more popularity this year. In fact, Google believes MPEG-DASH is the most valuable alternative for adaptive streaming [2]. The standard is quite similar to 3GPPs solution with some additions included. The media format is an extension to ISO Base Media File Format (ISOBMFF) or MPEG-2 Transport Stream which is different from 3GPPs proposal. The manifest format is similar to 3GPP called Media Presentation Description (MPD), a XML document that contains information where segments are located. One of the advantages in DASH is its support for novel video codecs as is stated in the following main feature list [44]

- Support for Scalable Video Coding(Scalable Video Coding (SVC)) and Multiview Video Coding (Multiview Video Coding (MVC))
- Efficient use of CDNs, proxies and NATs
- Entire stream control by the client
- Seamless switching of tracks
- Byte range requests
- Support for various segment formats, easy integration basically for any existing video codec
- Live, on-demand and time shift streaming services
- The concept of selectable and switchable streams

DASH standard [74] has already aroused interest among research institutes and companies mainly due to its possibility for adaptive streaming [54] [42]. For example, in [54] writers aim at finding good adaptation algorithm by dynamically selecting the best representation based on its average bitrate. Naturally the main goal in adaptive streaming is to provide good quality of experience for the end user. In [54] several other optimisation goals are in the target that can be generalise to adaptive HTTP streaming via DASH:

- Avoid interruptions of the playback due to buffer underruns

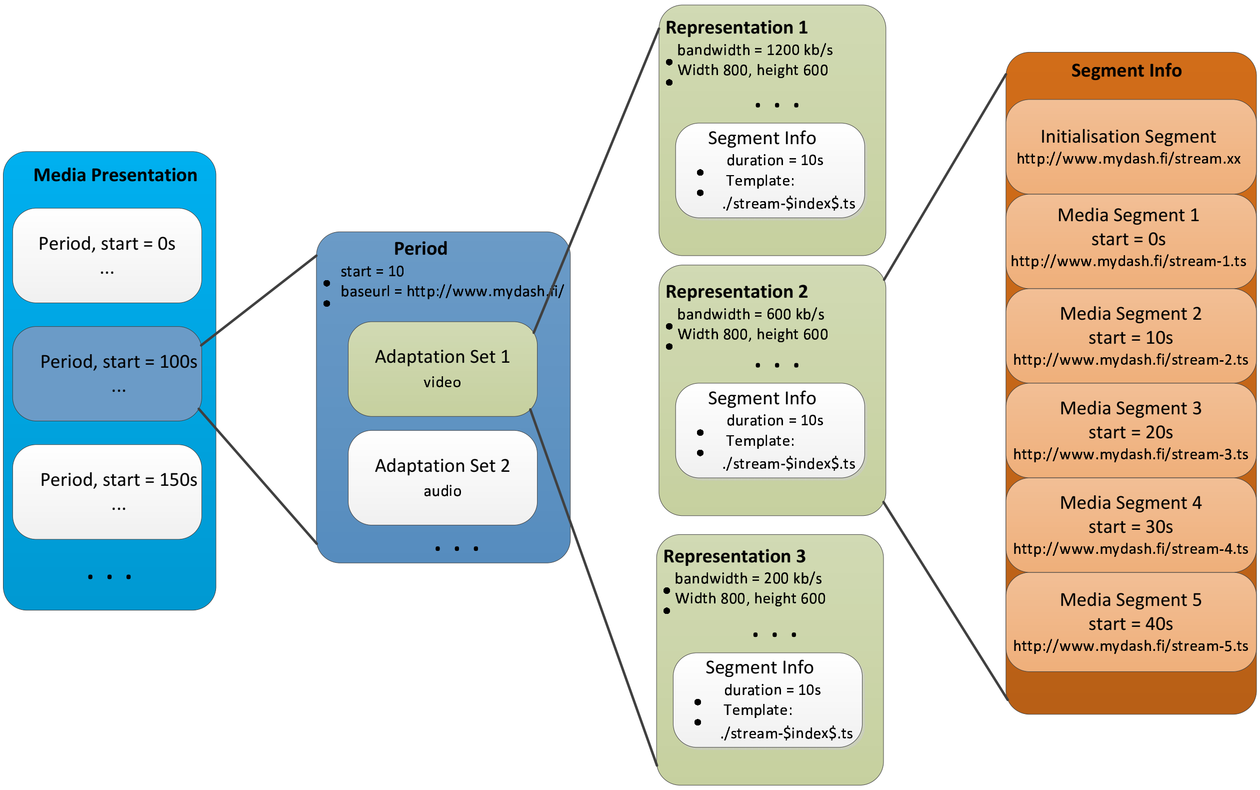


Figure 2: Illustration of Media Presentation Description(MPD).

- Maximise the minimum and average video quality
- Minimise the number of quality shifts
- Minimise the time after the user requests to view the video and before the start of the playback

MPD is a structured collection of media content in XML-format allowing the client to identify where video segments are located. In other words, client uses MPD to find the location and timing information in order to request and playback the content. MPD file is requested from the server in the beginning of the streaming. However, it can be updated during the session if needed.

MPD consists of three main components as can be seen in Fig. 2: period, representation and segment. Period component usually defines the adaptation sets and base Uniform Resource Locators (Uniform Resource Locator (URL)). In proportion, the representation component contains the information of each video's characteristics, such as bandwidth requirement (bitrate), segment duration and video/representation resolution. In other words, the representation component presents all the available encoded videos with variable characteristics, such as lower bitrate. The most important component is the segments which hold few seconds long portions of the actual content. Each individual segment can be presented individually without context to previous or following segments by using HTTP URLs. Segments can be also indexed to sub-segments and fragments.

2.2.4 HTTP Live Streaming

Apple's HTTP Live Streaming (HLS) [4] is a proprietary solution assessing to solve the adaptive HTTP streaming issue by using Apple's equipment as the end terminal. This solution has been introduced already in 2009. At the moment, HLS works in iPhone, iPad, iPod Touch, Android v.3.0 Honeycomb, VLC media player v.2.0, ffplay and

Quicktime player v.10+, just to name a few. The media format used in HLS is MPEG-2 Transport Stream(.ts) or MPEG-2 audio elementary stream. In contrast XML-based manifest file used e.g. in DASH, the manifest file format is M3U playlist.

HLS works like all the adaptive HTTP streaming solutions; multiple files are created in the player, which can adaptively change streams to optimize the playback experience. As a HTTP-based technology, no streaming server is required, which mean that all the switching logic resides on the player. In other words, the source video data is encoded into multiple files at different data rates, which are divided into short segments, usually between 5-10 seconds long. These are loaded onto an HTTP server along with a text-based .m3u8 extended manifest file.

2.2.5 HTTP Dynamic Streaming

Adobe has also HTTP streaming solution called HTTP Dynamic Streaming (HTTP Dynamic Streaming (HDS)) [3]. Adobe's solution works only with Adobe Flash Player 10.1 or later in MAC OS, Linux and Windows environments. Adobe uses F4F file as media format for the content and F4M manifest file format, basically similar XML-file as in other HTTP streaming solutions presented earlier. One limitation in HDS at the moment is its video codec support, only H.264 and VP6 are supported. The feature list in Adobe's solution include issues such as

- Robust and scalable delivery to various end devices with high quality
- Adaptive bitrate delivery by detecting client's bandwidth and computer resources
- Live or on-demand streaming support
- HD quality up to 1080p from 700kb/s to over 6Mb/s
- Adobe Access Server support for protected streaming by content encryption
- Open source, customisable media player framework (OSMF) in order to add support for different media players

2.3 Rate Control Algorithms

Adaptive streaming over HTTP is a challenging new technology and it is not yet fully clear how well does the technology perform under dynamic networks conditions. Especially the complex interaction between TCP congestion control and rate adaptation/control implemented on the application layer could be a challenge due to a possible complex feedback loop. This chapter focuses on rate adaptation (control) mechanisms at the application layer.

The main challenge in rate adaptation (control) mechanisms is to identify the available network resources and to detect congestion in the network early enough. The problem in the case of adaptive HTTP streaming is to differentiate between variations caused by TCP congestion control (usually short-term variations) and more persistent bandwidth changes caused by some other sources (such as heavy traffic, network device malfunction etc.). The problem is even worse in wireless networks where e.g. fading and shadowing may occur.

The purpose of the rate adaptation mechanisms in adaptive HTTP streaming is to choose segments of the representation that would allow user to receive smooth playback of the video under varying network conditions. Apart from the changes in the bandwidth availability, the rate-adaptation algorithm must compensate for the behaviour of the TCP congestion control which results in a saw-tooth shaped instantaneous transmission rate [39].

Using client-side buffering allows smoothing out the bandwidth variations. A number of segments can be buffered by the media client before being consumed by the media engine (i.e. decoder), thereby compensating for drops in the network throughput. However, buffer underflow and overflow should be handled carefully. In the event of buffer underflow, the user will experience interruptions in the playback, whereas in the event of overflow the rate-adaptation algorithm must make sure that the user receives video with the best quality (i.e. at highest possible bit-rate) and there is not underutilisation of the network. The algorithm must also take into account that frequent jumps between high and low bit-rates can be more frustrating for users than staying at a low quality for a substantial amount of time.The buffering in adaptive-streaming solutions depends on two factors: i) the buffering space on the media client and ii) the start-up delay requirement, i.e. the amount of time a user has to wait from the moment a video is requested until the video playback starts. Although a delay of up to 30 seconds is typical for non-commercial systems, the 2-4 seconds requirement is still a target for commercial deployments for Internet TV and IPTV. The set of decisions that can be taken by the rate-adaptation algorithm corresponds to the set of representations available at the web server. Even though the outcome of the decision is not completely under control of the media client due to the influence of stochastic variables (e.g. network throughput and segment size), the decision should be based on a certain strategy that takes into account how the planned action contributes to the overall quality of experience (Quality-of-Experience (QoE)). For example, the algorithm may choose a representation that is likely to have a long transmission time (thus, increasing the risk of buffer underflow), but that produces a video of higher quality. On the other hand, the algorithm may choose a presentation with lower quality predicting that it would lead to higher QoE (e.g. by minimizing the risk of buffer underflow) in the long run. These considerations should be taken into account when designing the rate adaptation algorithm.

2.3.1 Rate-adaptation algorithm - MS Smooth Streaming

Smooth Streaming is a proprietary technology introduced by Microsoft and it is very similar to generic adaptive HTTP streaming technology introduced earlier in this document. Similar to other adaptive HTTP streaming technologies, Smooth Streaming is based on manifest file(s) and segmented media content. Smooth Streaming is using two TCP connections to transfer the data from the server to the user. The first one is used to transfer audio segments and the other one is used to transfer video segments. However, under certain conditions both connections can be used to transfer video segments [9].

Smooth Streaming implementation available through Microsoft Silverlight is operating in two states: Buffering and Steady-State and operation is a bit different in these states. In a buffering state the player requests a new segment as soon as the previous one has been downloaded in order to fill the buffers as soon as possible. In Steady-State the player requests a new segment when needed to maintain a constant playback buffer. The adaptation strategy of Smooth Streaming client is conservative which means that it prefers to maintain a gap between the available bandwidth and the requested bit rate and not to try to utilise the full bandwidth. The strategy also includes faster quality transitions to higher quality than transitions to a lower quality and it also avoids large transitions in quality (bitrate). The experiment done in [9] indicates that the client does the adaptation according a smoothed estimate of the bandwidth and not the latest fragment download throughput.

2.3.2 AdapTech Streaming algorithm

Authors of [9] propose a rate-control algorithm for adaptive HTTP streaming which is based on Smooth Streaming algorithms but it tries to address the weaknesses of Smooth Streaming clients presented earlier. The design objectives of the proposed algorithm, presented below, are actually a good list of design principle for every adaptive HTTP streaming rate-control algorithm.

- Start the video playback quickly as possible which means that the initial playback delay should be much shorter than the maximum delay in the playback buffer
- Avoiding playback buffer underflows and video freezes should be the top-priority of the algorithm
- Smooth transitions between different representations (bitrates) so that the end-user does not notice large quality changes
- Detect stable bandwidth increases and switch to a higher content representation
- Do not react to short-term bandwidth fluctuations

The proposed algorithm has two states similar to Smooth Streaming clients: Buffering and Steady-State. In the buffering state the player tries to fill the buffer as fast as possible and it switches to the Stead-State when the buffer is full enough (predefined value). However, the player can start playback when the buffer status reaches a predefined threshold. The algorithm makes the decision to switch to another profile after every downloaded segment based on two metrics, throughput of the last downloaded segment and smoothed average of earlier downloaded segments. The logic behind the algorithm can be determined as follows:

```
if $t2 < bw(t)$ then
    if $pr2 > prcurr$ AND switch-up flag is TRUE then
        increase the requested bitrate by one profile
    end if
else if $t1 < bw(t) < t2$ then
    if $pr1 > prcurr$ then
        decrease the requested bitrate by one profile
    else if $pr1 > prcurr$ then
```

```
        increase the requested bitrate by one profile
    end if
else if $bw(t) < t1$ then
    switch to the lowest available profile
end if
```

Where $t1$ and $t2$ are thresholds for buffer fullness, $bw(t)$ is the buffer size at time instant t. $pr1$ and $pr2$ are candidate bitrate profiles and $prcurr$ is the current bitrate profile. The proposed algorithm seems to achieve the target design goals of the algorithm pretty well and it adapts to bandwidth fluctuations better than the commercial Smooth Streaming algorithms.

2.3.3 Multilink-based approach for rate-control

The approach proposed by [22] is based on multilink retrieval of media segments and the most important part of the controlling algorithm is the request scheduler which controls when request for new segments are made and also controlling the video quality (bit rate).The importance of controlling the request process of segments is emphasised when multilink retrieval is utilised. The algorithm is using throughput and Round-Trip Time (RTT) to evaluate the link characteristics. The algorithm calculates how much content it has received and is ready for playout. It estimates how long it takes to receive already requested data. This is subtracted from the amount of data already downloaded to get an estimate of how long the client can spend receiving new segment without causing a deadline miss. This estimate is compared against estimates of the time it takes to receive the segment at different bit rates and the most suitable version of the segment is selected. The authors have proposed that introducing multilink approach together with dynamic subsegment request approach do improve the system in terms of improved video quality and smaller receive buffer.

2.3.4 Rate-control algorithm with improved QoE and bandwidth estimation

One of the most important metric for the algorithm is to evaluate the available bandwidth as correctly as possible. Authors of [55] propose a QoE-aware DASH system which has a separate measurement proxy implemented immediately next to the video server for this purpose. The measurement proxy manipulates the video data packets by introducing inline measurement information into the packets. This architecture removes the need to perform the measurement at the client side or at the server side and introducing the probing information inline the need for separate measurement packets common for bandwidth probing. However, the proposed architecture requires modifications on the application side and it requires an additional proxy to be included in the network. In the proposed architecture the actual rate-control algorithm is implemented at the client side and it relies on bandwidth estimations achieved through the measurement proxy. The main idea behind the proposed algorithm is to allow more QoE-aware transitions between different representations of the content when the network gets congested.

2.3.5 Server-based rate-control algorithm

Authors of [16] propose a rate-control algorithm which relies on server-based rate-control algorithm. Compared to other introduced algorithms, this algorithm is implemented at the server-side in order to reduce the delay and remove the need of explicit feedback from the client. The algorithm is based on controlling theory (PI-controller) and it is controlling the sender buffer fullness at the server side.

As a conclusion from different rate-adaptation algorithms, it can be seen that designing an efficient rate-adaptation algorithm for adaptive HTTP streaming functioning at the top of the TCP protocol is not an easy task. It is not yet well understood how two competing controlling loops (rate adaptation at the application layer and TCP congestion control) work together and especially when several competing clients are acting over the same network link [8]. Also the importance of accurate bandwidth estimation should be emphasized since it is the most important metric available for the rate-control algorithms.

2.4 Cross-layer prioritization technique for HTTP streaming

2.4.1 Context of the study

Through this study, we wanted to raise a typical issue linked to the use of HARQ retransmissions in a situation of mobility or in a very degraded channel context. In a context of mobility, with a varying channel, retransmissions linked to the HARQ process may generate disorders and delays among packets in reception. Whereas such disorders will be masked with TCP protocol or just generate more delays, when the real time transport protocol (RTP) protocol is used for streaming, packets disordered and delayed are very prejudicial to the resulting quality of the video in reception. Actually, the RTP protocol does not need acknowledgments. It means that new packets may be retransmitted while new ones from User Datagram Protocol (UDP) may be sent for the first time. Once arrived at the application layer, disordered packets are rejected.
Hence the different strategies proposed to struggle against disordered packets in reception or in emission, with a RTP transport layer :

- Packets may be sequenced at the RTP layer. However, it may generate large delays and thus, be contrary to a real-time display of the video.
- Another strategy would schedule packets according to the importance of frames in emission. Indeed, some frames do not need to be retransmitted or even transmitted. When one of this frame is missing, the resulting quality of the video is not so damaged. Thus, avoiding the transmission of such frames may bring improvements.

The adopted solution is a mix of the propositions mentioned above added to an adaptation on channel indications.

2.4.2 Main idea and principles of the algorithm proposed

Disordering linked to HARQ use In that section, the HARQ mechanism chosen is the parallel Stop and Wait one. Thanks to this technique, several MAC packet units may be sent at the same time, avoiding loosing delay between transmissions. However, in the context of a very varying channel, one or several MAC packet units composing a slot may be lost. These latter will then be retransmitted provided their own remaining persistency is high enough (the persistency could be understood as a maximum credit of retransmissions for a particular MAC packet unit). As shown on two HARQ exchanges for sake if simplicity in figure 3, this mechanism could lead to packet disordering in reception.

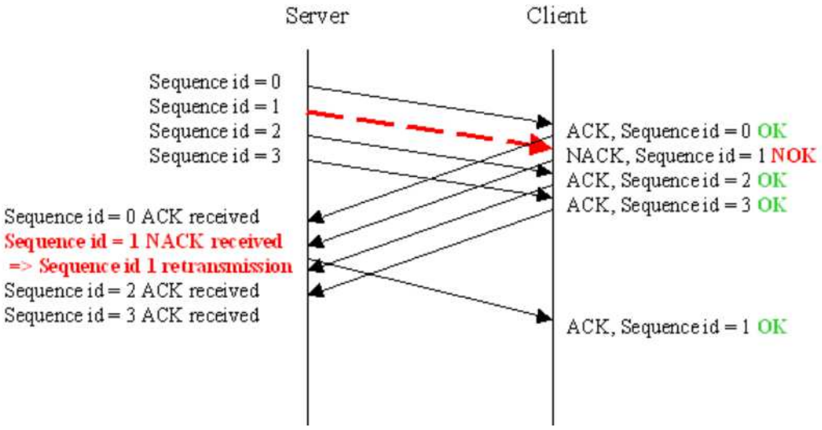

Server
Client
Sequence id = 0
Sequence id = 1
Sequence id = 2
Sequence id = 3
ACK, Sequence id = 0 OK
NACK, Sequence id = 1 NOK
ACK, Sequence id = 2 OK
ACK, Sequence id = 3 OK
Sequence id = 0 ACK received
Sequence id = 1 NACK received
=> Sequence id 1 retransmission
Sequence id = 2 ACK received
Sequence id = 3 ACK received
ACK, Sequence id = 1 OK

Figure 3: Example of disordering resulting from the HARQ technique

Error correcting codes used Low-density parity-check (LDPC) codes, matching with the 802.16e (coding rate $1/2$, code type: Structured IRA codes [86]) specifications were simulated in order to acquire their performances for several successive retransmissions, in terms of PER (figure 4). Chase combining was used on the LLRs obtained for each transmission to cumulate the effect of repetitive transmissions.

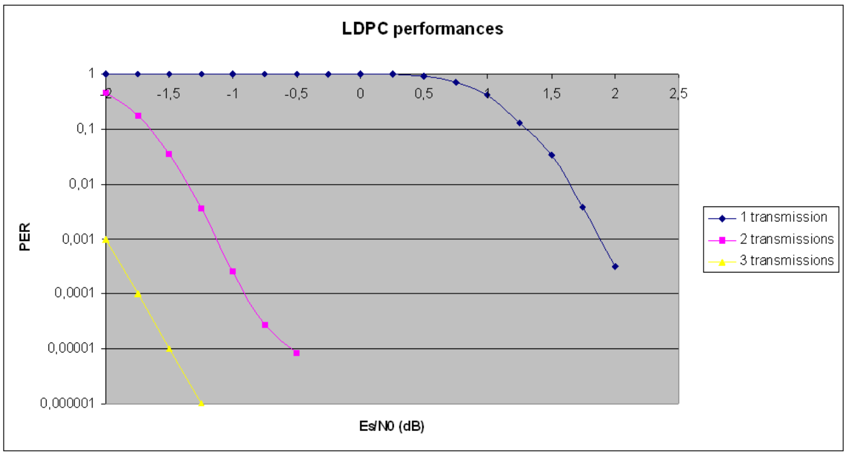


Figure 4: LDPC performances for 1, 2 and 3 cumulative transmissions

A packet reordering technique in the transmission part The main idea of this HARQ based technique is to sort packets from the transmission part in order to avoid long delays due to reordering on Datalink or Transport Layers. In the following, this technique will be called “Adaptive HARQ-based solution”. Contrary to a classical HARQ technique, Adaptive HARQ-based solution tries to limit delay by planning the future. When a NACK is received (this NACK contains an information on the channel status), the Adaptive HARQ-based solution refers to the LDPC performances presented above to estimate the necessary number of packet retransmissions to plan in order to match with a fixed desired PER threshold. Knowing this parameter, the necessary number of retransmissions is directly sent to the client. Thus, there is no need anymore to wait for several NACKs. It assumes that the channel is stationary enough in that period. The Adaptive HARQ-based solution is illustrated in figure 5, where a maximum persistency of 3 transmissions is considered.

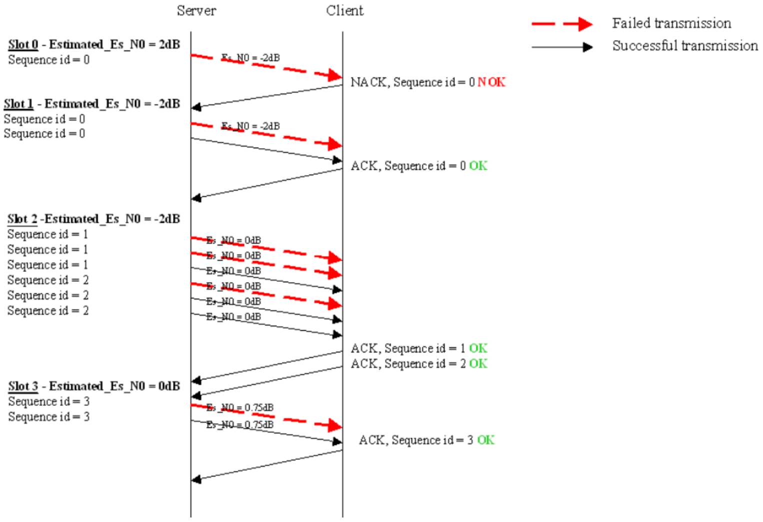


Figure 5: Adaptive HARQ mechanism

2.4.3 Assumptions on the simulations realized

Definitions of the metrics used

Referring to the literature [7] and [64], some metrics were adopted to qualify packet reordering or packet disorders.

Reorder density is the distribution of displacements of packets from their original positions, normalized with respect to the number of packets. Thus, an early packet goes with a negative displacement and a late packet with a positive one. A threshold value on displacement may be defined. That way, a threshold to define the condition under which a packet is considered as lost may be useful.

Reorder buffer-occupancy density is defined as the normalized histogram of the occupancy of a hypothetical buffer that would allow the recovery from out-of-order delivery of packets. Thus, if a packet with a sequence number greater than expected arrives, it is considered to be stored in the hypothetical buffer until it can be released in order. The occupancy of this buffer, after each arrival, is used as a measure of reordering.

Architecture of the Omnet++ software

The figure 6 presents the architecture currently implemented to simulate the protocol stacks designed for the study. As a reminder, a simulation on OMNet++ may be considered as composed of entities of two natures. First of all, the nodes of the network: each of them contains an instance of the set of protocols to simulate. Each instance has its own data , and among the set of protocols, each entity communicates through the lower layers (until the Physical Layer). Then, some services allowed by the simulation environment are shared between nodes.Each module in the simulation environment is defined by a NED file, i.e. a text file using the OMNeT++ NED syntax. This file defines a complex OMNeT++ module in which all protocol layers are mentioned.

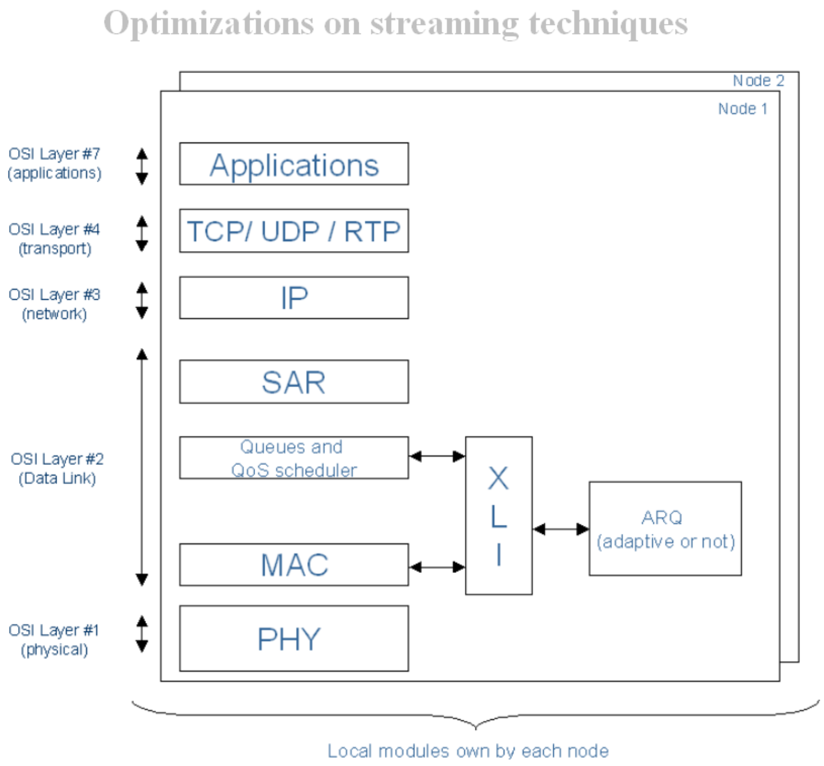


Figure 6: Simulation protocol architecture

The simulation environment is composed of different services shared by each node which are:

- A log service that allows to keep information on key points of the simulation
- A way to exchange data messages between layers
- A cross-layer interface allowing signalling communications

Data exchanges between layers To allow the transmission of data between layers, OMNeT++ uses messages, which are C++ objects derived from **cMessage** class. During the transmission period, each protocol layer of level n-1 allocates its own message, which encapsulates the precedent message from the higher layer n. During the reception period, each protocol layer of level n receives its information and decapsulated the message of level n+1 from the message of level n, before transmitting it to the upper layer, n+1 leveled.
To improve precedent way to transmit messages, no more encapsulation is done by the different messages and a memory area is associated to each data message. This memory area is large enough to host the information of the layer generating this message, but also, to receive all the possible headers and data added by the lower layers. Thus, the same **cMessage** object is transferred all along the different protocol layers. The derived class from **cMessage**, **BytesMsg** has three specific members :

1. *memoryArea*, is a pointer to the data area where the bytes of the message are stored;
2. *memoryAreaBytes* memorizes the whole memory size;
3. *pduBytes* memorizes the actual length of the message. The first byte of the message has the following address, illustrated by , *memoryArea-(memoryAreaBytes+pduBytes)*.

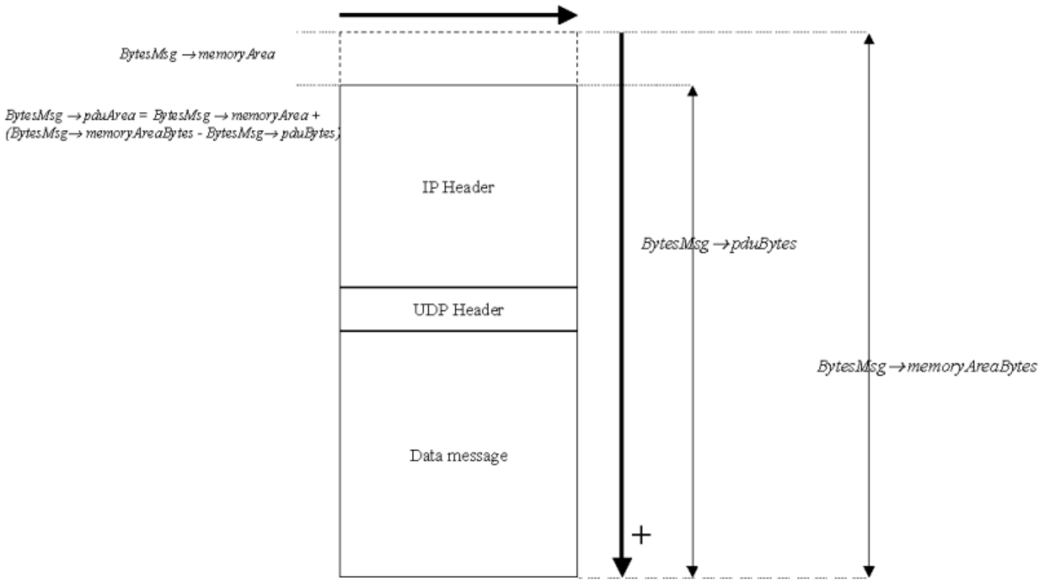

BytesMsg → memoryArea
BytesMsg → pduArea = BytesMsg → memoryArea +
(BytesMsg→ memoryAreaBytes - BytesMsg→ pduBytes)
IP Header
UDP Header
Data message
+
BytesMsg → pduBytes
BytesMsg → memoryAreaBytes

Figure 7: Data message architecture

During a reception phase from a higher layer, a protocol entity adds its own header before the first byte of the message received (beginning of pduArea), raises the data member *pduBytes* of the header size, and forwards message to the lower layer. During a reception from a lower layer, the protocol entity reads the header inserted by its homologous entity at the transmitter side, reduces *pduBytes* by the header size and forwards message to upper layer. With such a mechanism, we do not need to use specific classes derived from **cMessage** for each particular protocol level. A generic class is only used : **BytesMsg** (figure 7).

Cross layer interface for signalization A XLI module is implemented to link some layers together. To allow such a communication, each layer concerned has its own gates (input and output) to receive and transmit controlling information from the XLI. Each controlling message transports identification information to know which module must receive the message.

Simulation framework All entities forming the protocol stack compose the simulation framework. The main components of the structure are :

- a traffic generator RTP/UDP and TCP as well,
- an IP layer divided into two sub-layers
- a SAR layer
- a layer taking into account the queues and the QoS scheduler algorithm
- a simplified MAC layer allowing different possible modes of retransmissions thanks to a specified module,
- a simplified Physical layer taking into account a complex signal and different types of channels : additive white Gaussian noise (AWGN) and Gilbert-Elliott

Cross layer signaling information Different controlling information is attached to messages during the simulation. A *TransportControlInfo* object is attached to each application message to be transmitted with UDP. A *Layer2ControlInfo* object is attached to messages during the transfers of messages of DataLink layer. Similarly, a *Layer1ControlInfo* is linked to messages when they are transmitted through the Physical Layer. It is filled with lots of information, not always necessary but useful to check the validity of the code and write it in logs. Among them, are :

- source IP address,
- destination IP address,
- source port,
- destination port,
- priority,
- an indicator to know if RTP is encapsulated,
- an indicator to know which IP version is chosen,
- flow label,
- *udpPacketLength*,
- checksum emitted and written in the header,
- *nbFilesPlayed* for video payloads,
- traffic class,
- *udpIdentifier*.

The transmission of the **BytesMsg** and their attached *ControlInfo* is illustrated by figure 8.

HARQ and adaptive HARQ implementation The different steps illustrated in figure 9 show how HARQ and adaptive HARQ were implemented.

Step 1: each time a data slot must be sent, the ***MAC module*** requests packets to the ***FifoQueues*** in order to fill in the data slot and transmit it to lower layers. Retransmissions are privileged in comparison with first transmissions. For the adaptive HARQ, the same packet may be repeated. This repetition depends on the Es/N0 value returned from the channel.

Step 2: the ***MAC module*** sends packets to the ***RetransmissionMechanism module***. The latter memorizes the transmitted packet, updates its persistency and sends back packets to the ***MAC module***.

Step 3: once received by the ***MAC module***, packets will be forwarded to the ***TSM*** (Transmission Scheme Manager) ***module***. This module plays the role of scheduler between the different modules of the physical layer : ***forward error correction (FEC), Modem***...

Step 4: Packets are encoded by the ***FEC module***, according to the type of codes desired for the simulation assumptions.

Step 5: Packets are modulated.

Step 6: The ***TSM*** sends the complex signal generated through an ***AWGN channel***.

Step 7: The receiver receives packets sent by the server and forwards them to the ***Modem*** via the TSM.

Step 8: Packets are demodulated.

Step 9: Packets are decoded:

- If a packet is correctly decoded, an ACK is sent back to the server to inform it that no retransmission is necessary for this particular sequence number. The packet is forwarded to upper layers. The information on the Es/N0 value should also be transmitted at this moment. For sake of simplicity

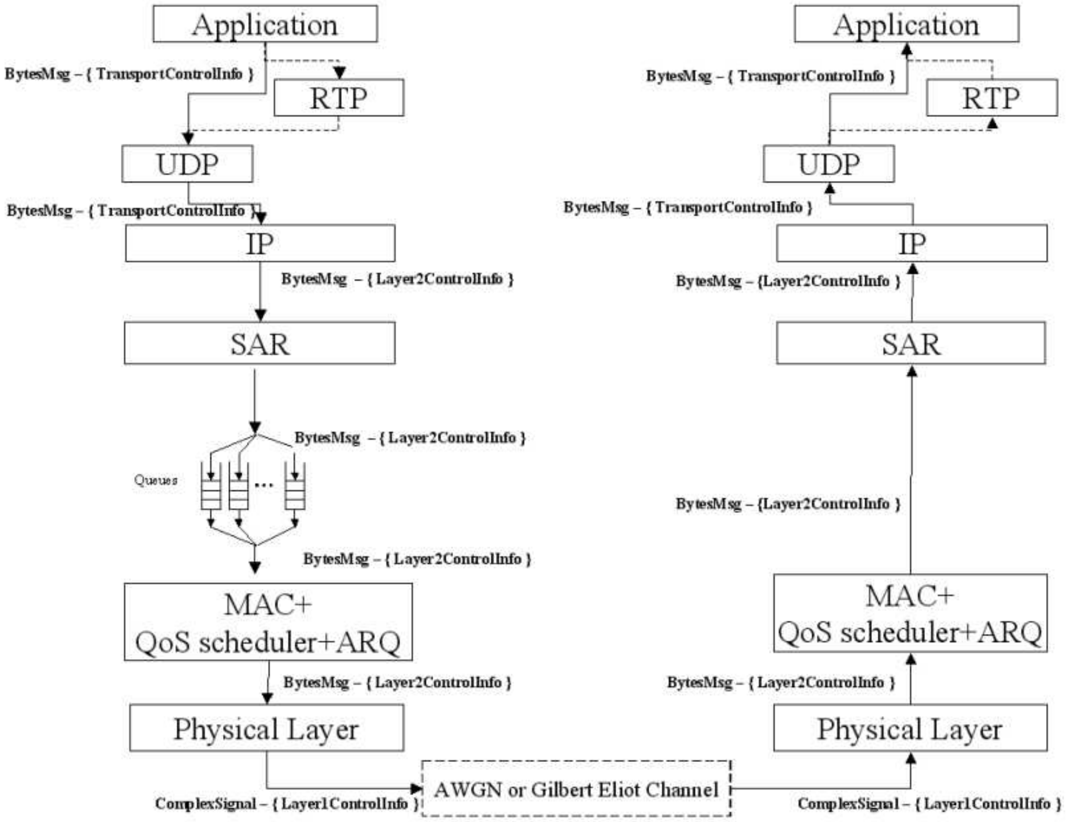

Application
BytesMsg – { TransportControlInfo }
RTP
UDP
BytesMsg – { TransportControlInfo }
IP
BytesMsg – { Layer2ControlInfo }
SAR
BytesMsg – { Layer2ControlInfo }
Queues
...
BytesMsg – { Layer2ControlInfo }
MAC+
QoS scheduler+ARQ
BytesMsg – { Layer2ControlInfo }
Physical Layer
ComplexSignal – { Layer1ControlInfo }
AWGN or Gilbert Eliot Channel
Application
BytesMsg – { TransportControlInfo }
RTP
UDP
BytesMsg – { TransportControlInfo }
IP
BytesMsg – {Layer2ControlInfo }
SAR
BytesMsg – {Layer2ControlInfo }
MAC+
QoS scheduler+ARQ
BytesMsg – { Layer2ControlInfo }
Physical Layer
ComplexSignal – { Layer1ControlInfo }

Figure 8: Data message passing architecture

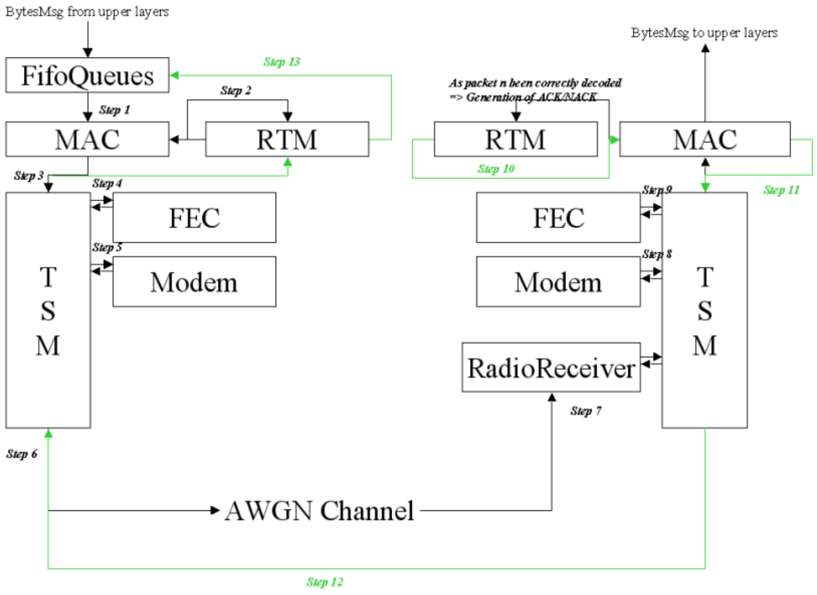


Figure 9: HARQ implementation architecture (in green, the different steps regarding ACK/NACK transmissions)

for the implementation model, instead of transmitting it, the Es/N0 value is kept in memory in a EsN0.txt file and used two slots later.

- If a packet is not decoded, its LLRs are saved to be combined later, a NACK is transmitted to ask for retransmissions. The packet is deleted.

Step 10:the RTM generates a ACK or a NACK according to the successful decoding of a packet i.

Step 11: ACK or NACK packets are sent on the next data slot corresponding to the data transmission of the client node.

Step 12:Packets are directly sent on a perfect channel to the server node.

Step 13:Packets are interpreted by the RTM module. For a NACK, if the persistency credit is high enough, it asks for retransmissions of packet n, to the ***FifoQueues***. Otherwise, if the acknowledgment received is either an ACK or if no more retransmission credit is available, savings related to packet n are deleted.

2.4.4 Performance and interpretation

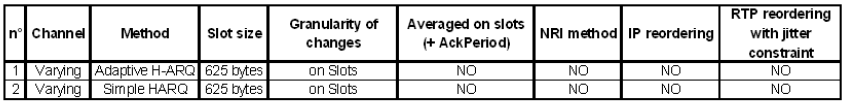
In this part, the scenarios tested will be listed to highlight the benefits of the adaptive algorithm used on HARQ. As this technique competes with a simple HARQ with reordering in reception, a fair way to determine the real gains of the adaptive HARQ will be to compare this two techniques. That is why, two treatments were developed in the simulator in order to reorder packets in the transmission part of the architecture, as well as in the reception part.
Two different channel models will be adopted to compare the behaviors of the two versions of HARQ :

- a very varying channel with Eb/N0 values chosen between $-2\ dB$ and $2\ dB$, with a $0.25\ dB$ step
- a Gilbert-Elliot channel with transitions between $-2\ dB$ and $2\ dB$ states

The different scenarios are listed below :

Tests 1 and 2

Packets are sent on a $1Mbit/s$ link with varying perturbations on channel, ranging from $-2\ dB$ and $2\ dB$.



n°	Channel	Method	Slot size	Granularity of changes	Averaged on slots (+ AckPeriod)	NRI method	IP reordering	RTP reordering with jitter constraint
1	Varying	Adaptive H-ARQ	625 bytes	on Slots	NO	NO	NO	NO
2	Varying	Simple HARQ	625 bytes	on Slots	NO	NO	NO	NO

Table 1: Assumptions on tests 1 and 2

According to figure 10 and figure 11, the probability density function attached to a null displacement is higher for the Adaptive HARQ based solution than for the classical HARQ method. It means that less disorders are brought by the new technique we are proposing.

As regards the PSNR evolution, as shown in figure 12, the Adaptive HARQ-based solution is far more effective than the HARQ, even though some periods are impacted by failures on the video. It means that the adaptive HARQ was not able to prevent the bad conditions on the channel for that late periods.

Tests 1, 1_1, 1_2 and 1_3

These tests are performed on a $4Mbit/s$ perturbated link (table 2). The estimation of Es/N0 is either estimated on the previous slot, either on the average value of Es/N0 on the two previous slots or on the average value of Es/N0 on the four previous slots.

Referring to figure 13 and figure 14, the probability density of null displacement keeps growing as the Es/N0 is estimated on larger ranges. Thus, an ideal method would be to average Es/N0 values on several slots to allow a smoother adaptation to channel conditions.

As illustrated by figure 15 and table 3, a greater throughput allows better performances in terms of delay, which is perfectly logical. As regards resulting PSNR values obtained in test 1, it is hard to compare fairly this PSNR results to the others since, the channel model, even identical seen by slot, does not impact the transmissions in the same way. Actually, we consider that the Es/N0 is seen by slot for sake of simplicity in the implementation.
However, in this set of tests, averaging Es/N0 values on at least 4 slots benefits to the Quality of Service provided to the client.

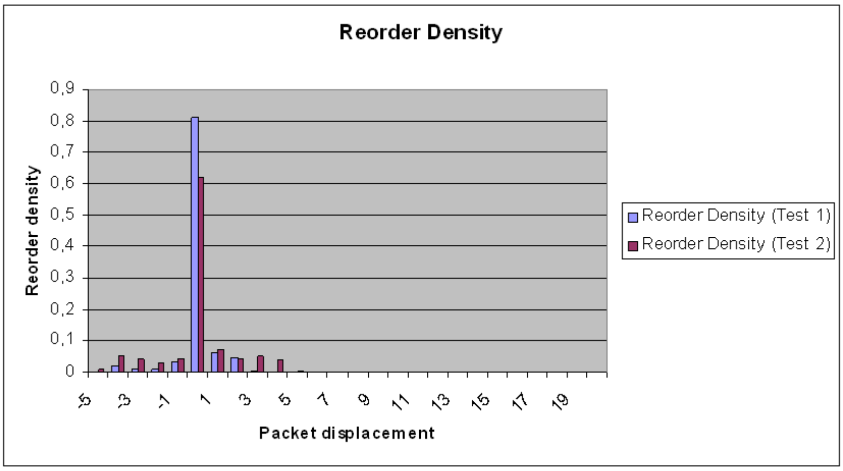


Figure 10: Reorder density distribution for scenarios 1 and 2

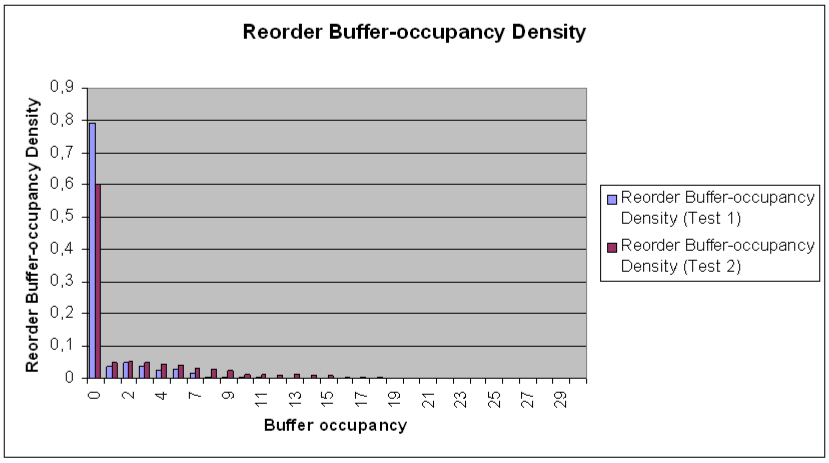
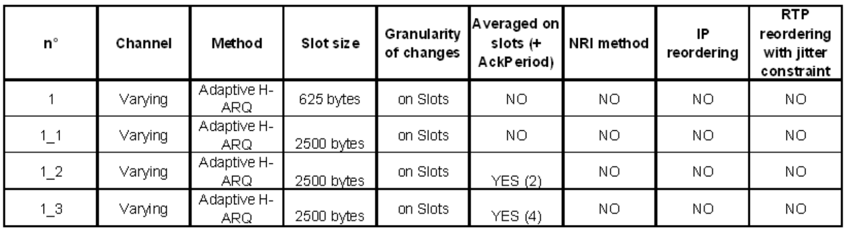


Figure 11: Reorder buffer-occupancy density distribution for scenarios 1 and 2



n°	Channel	Method	Slot size	Granularity of changes	Averaged on slots (+ AckPeriod)	NRI method	IP reordering	RTP reordering with jitter constraint
1	Varying	Adaptive H-ARQ	625 bytes	on Slots	NO	NO	NO	NO
1_1	Varying	Adaptive H-ARQ	2500 bytes	on Slots	NO	NO	NO	NO
1_2	Varying	Adaptive H-ARQ	2500 bytes	on Slots	YES (2)	NO	NO	NO
1_3	Varying	Adaptive H-ARQ	2500 bytes	on Slots	YES (4)	NO	NO	NO

Table 2: Assumptions on tests 1, 1_1, 1_2 and 1_3

	Test 1	Test 1_1	Test 1_2	Test 1_3
Average delay to receive an IP packet	0,058	0,009	0,009	0,008
PER	0	0	0	0
Average PSNR (dB)	37,925	38,300	38,284	38,311
Average PSNR MSE (dB)	35,204	37,988	37,98	38,118

Table 3: Statistics for Tests 1, 1_1, 1_2 and 1_3

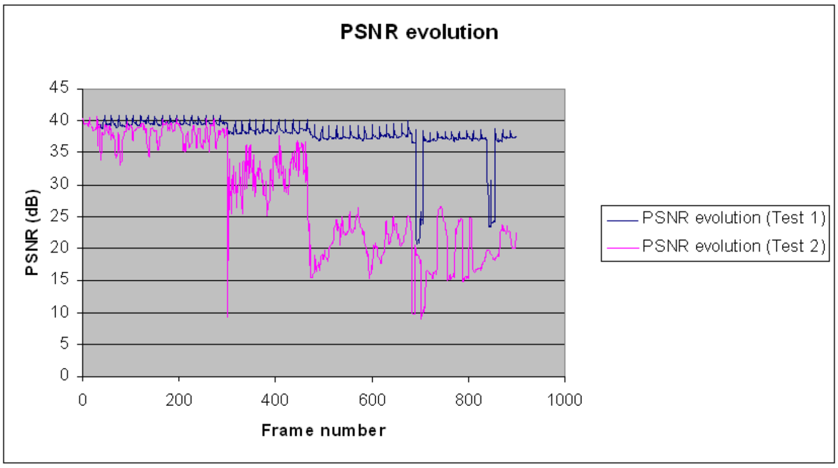


Figure 12: PSNR evolution for scenarios 1 and 2

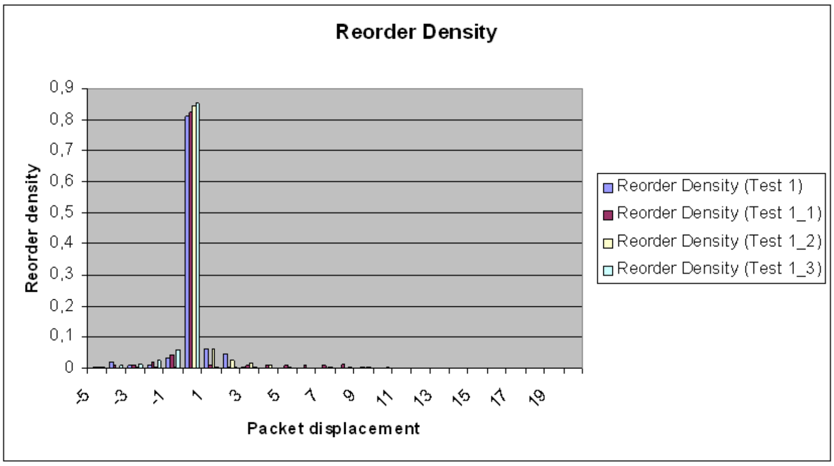


Figure 13: Reorder density distribution for scenarios 1, 1_1, 1_2 and 1_3

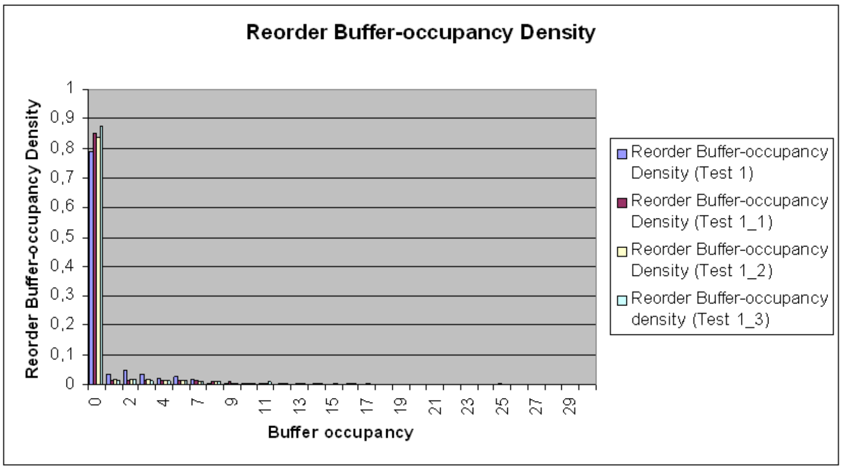


Figure 14: Reorder buffer-occupancy density distribution for scenarios 1, 1_1, 1_2 and 1_3

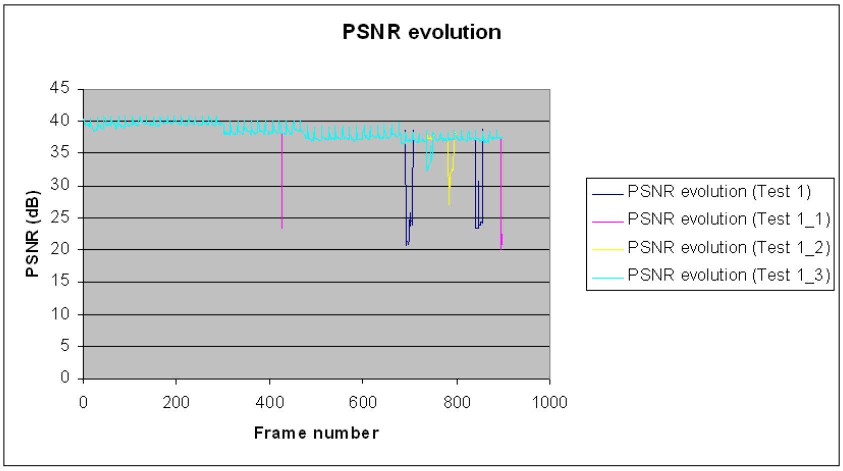


Figure 15: PSNR evolution for scenarios 1, 1_1, 1_2 and 1_3

Tests 1_1, 1_3 and 2_1

A comparison is performed between results from the Adaptive HARQ base solution and the HARQ solution, on a varying 4 Mbit/s link (table 4).

n°	Channel	Method	Slot size	Granularity of changes	Averaged on slots (+AckPeriod)	NRI method	IP reordering	RTP reordering with jitter constraint
1_1	Varying	Adaptive H-ARQ	2500 bytes	on Slots	NO	NO	NO	NO
1_3	Varying	Adaptive H-ARQ	2500 bytes	on Slots	YES (4)	NO	NO	NO
2_1	Varying	Simple HARQ	2500 bytes	on Slots	NO	NO	NO	NO

Table 4: Assumptions on tests 1_1, 1_3 and 2_1

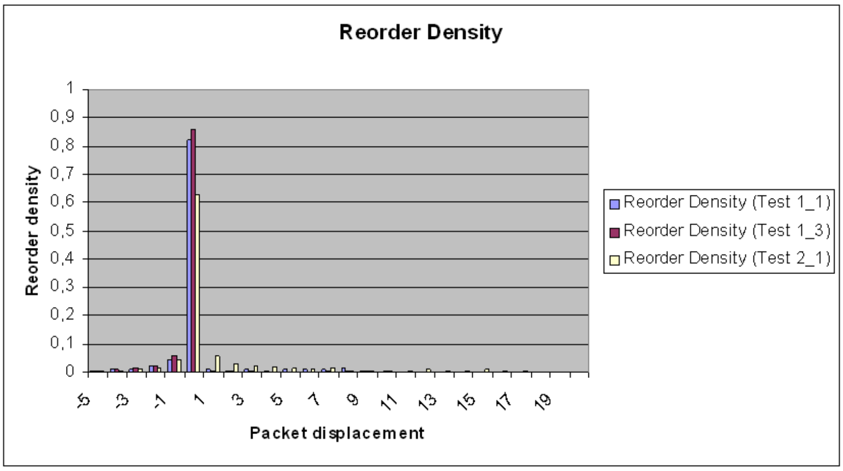


Figure 16: Reorder density distribution for scenarios 1_1, 1_3 and 2_1

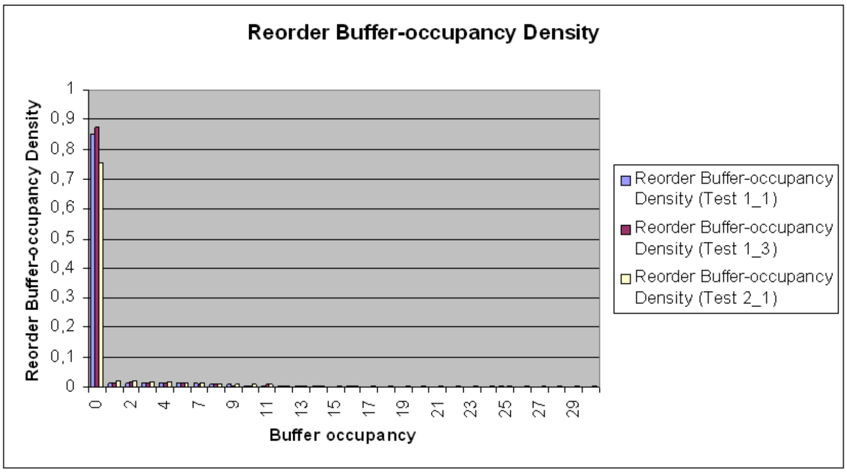


Figure 17: Reorder buffer-occupancy density distribution for scenarios 1_1, 1_3 and 2_1

Disordering is much more important using the simple HARQ technique. It may implicate larger delays than for Adaptive HARQ before reordering a whole image.

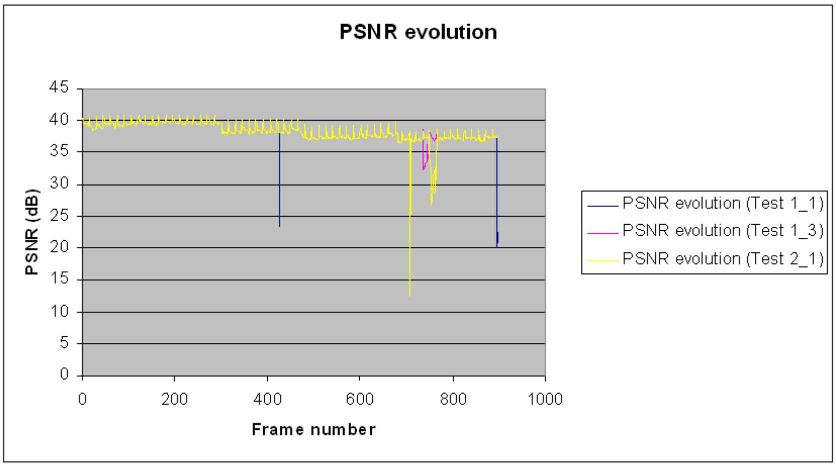
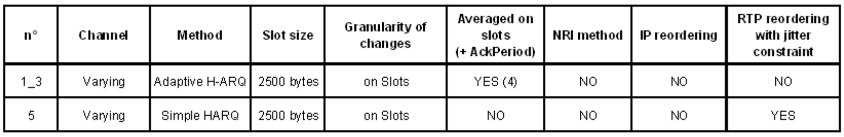


Figure 18: PSNR evolution for scenarios 1_1, 1_3 and 2_1

According to figure 18, the PSNR resulting from a transmission with the simple HARQ technique is more degraded than with the two methods using Adaptive HARQ-based solution. The PSNR level suddenly drops at the end of the video with the simple HARQ technique. It means that some resulting images were not correctly rebuilt : maybe disordered NALs were missing and avoiding the right reconstruction of images.
The average necessary time to receive an IP packet, in the context of a simple HARQ transmission and with this type of channel, is twice as high as the one resulting from the transmission with adaptive HARQ-based methods. Nevertheless, even though performances obtained thanks to adaptive HARQ-based methods are greater than for simple HARQ, it is not completely fair to compare these techniques, that way.
Actually, in a real context, RTP reordering would be used in reception, for HARQ transmissions, to privilege the QoS. The real challenge would be to show that adaptive HARQ method competes with HARQ with RTP reordering in reception.

Tests 1_3 and 5



n°	Channel	Method	Slot size	Granularity of changes	Averaged on slots (+ AckPeriod)	NRI method	IP reordering	RTP reordering with jitter constraint
1_3	Varying	Adaptive H-ARQ	2500 bytes	on Slots	YES (4)	NO	NO	NO
5	Varying	Simple HARQ	2500 bytes	on Slots	NO	NO	NO	YES

Table 5: Assumptions on tests 1_3 and 5

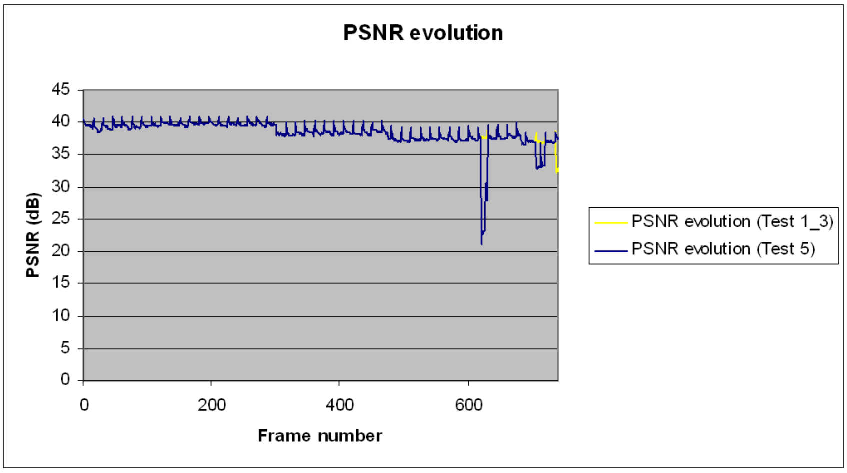


Figure 19: PSNR evolution for scenarios 1_3 and 5

This test aims at comparing the performance obtained between the previously explained test 1_3 and test 5, presented as the illustration of RTP reordering for a simple HARQ method (table 5). Disordering metrics are not compared for these tests as no disorder appears after RTP reordering. However, as the RTP reordering algorithm is simulated in the second test, some packets, arriving too late in comparison with the previous packets are discarded.

The Adaptive HARQ based method (with or without averaging estimations of Es/N0 on slots) is a perfect challenger with the RTP reordering method in reception, used on a traditional HARQ method (figure 19). Both methods have the same rough estimate of PSNR in reception. Nevertheless, if the reception time is taken into account, the adaptive HARQ based method emerges victorious from the comparison.

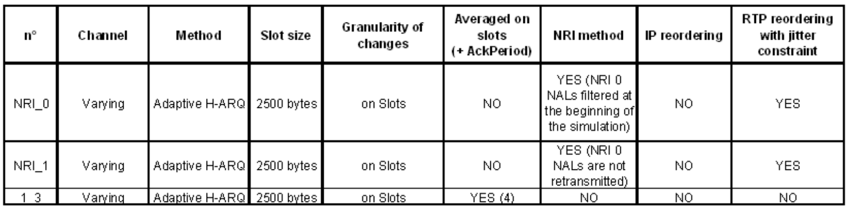
To be more competitive, the adaptive HARQ based solution could avoid exhaustive retransmissions of the packets composing a NRI-0 image. This technique would avoid wasting transmission resources considering that these packets are not necessary to guarantee a good PSNR value at the client side.

In the following test, we assumed that NRI-0 packets were given a lower retransmission credit.

Tests with NRI_0 filtering

The figure 20 illustrates the evolution of PSNR for three techniques, on the first 300 frames of the video ("MixSequenceQcif.264" which is a mix of several video references), to avoid the desynchronization due to the concatenation of small and very different videos:

- the first technique consists in filtering NRI_0 images at the beginning of the simulation. No transmission resource is wasted to transmit such images. The resulting average PSNR corresponding to that method is $35.885dB$.
- The second technique aims at not retransmitting NRI_0 packets when the channel is too degraded to allow the correct decoding of these packets, when they are sent for a first attempt. The resulting average PSNR corresponding to that method is $36.5443dB$.



n°	Channel	Method	Slot size	Granularity of changes	Averaged on slots (+ AckPeriod)	NRI method	IP reordering	RTP reordering with jitter constraint
NRI_0	Varying	Adaptive H-ARQ	2500 bytes	on Slots	NO	YES (NRI 0 NALs filtered at the beginning of the simulation)	NO	YES
NRI_1	Varying	Adaptive H-ARQ	2500 bytes	on Slots	NO	YES (NRI 0 NALs are not retransmitted)	NO	YES
1_3	Varying	Adaptive H-ARQ	2500 bytes	on Slots	YES (4)	NO	NO	NO

Table 6: Assumptions on tests NRI_0, NRI_1, 1_3

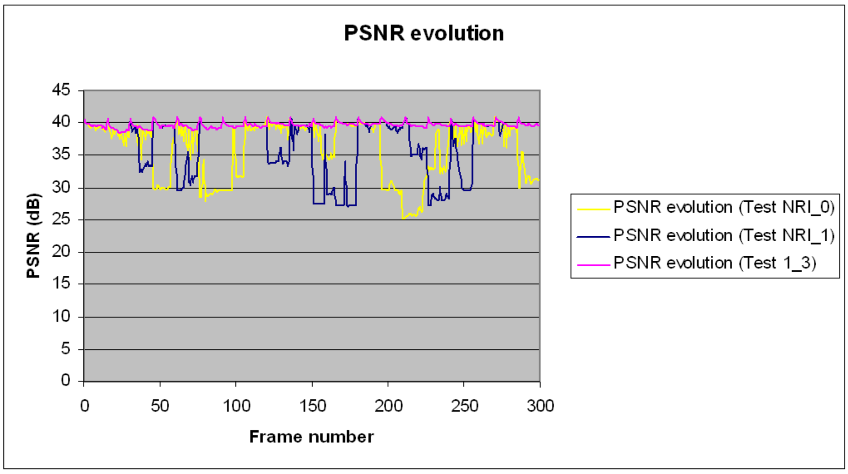
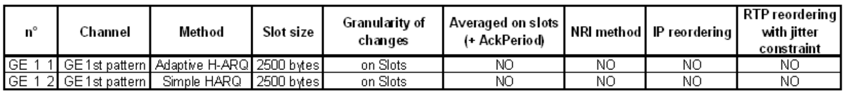


Figure 20: PSNR evolution for scenarios NRI_0, NRI_1 and Test 1_3

- The last technique is described above and is used as an illustration. The average PSNR calculated is equal to $39.65dB$.

As a conclusion for that set of simple tests (table 6), the transmission of NRI_0 should be considered as a lever between congestion avoidance and QoS. More simulations must be performed to validate these results, on other videos. Adaptive techniques based on this method could be proposed to regulate queuing congestion in transmission.

Tests GE_1_1 and GE_1_2



n°	Channel	Method	Slot size	Granularity of changes	Averaged on slots (+ AckPeriod)	NRI method	IP reordering	RTP reordering with jitter constraint
GE_1_1	GE 1st pattern	Adaptive H-ARQ	2500 bytes	on Slots	NO	NO	NO	NO
GE_1_2	GE 1st pattern	Simple HARQ	2500 bytes	on Slots	NO	NO	NO	NO

Table 7: Assumptions on tests GE_1_1 and GE_1_2

Some tests were led on less varying channels as Gilbert-Elliot models. The conclusions brought by such simulations were less impressing than for a worst-case, very varying channel. That is why the different results were not added to this document. In figure 21,22 and 23, some simple results on a Gilbert-Elliot channel ($pBG = 2,89.10-2$; $pGB = 2.91.10^{-3}$) are given for illustration.

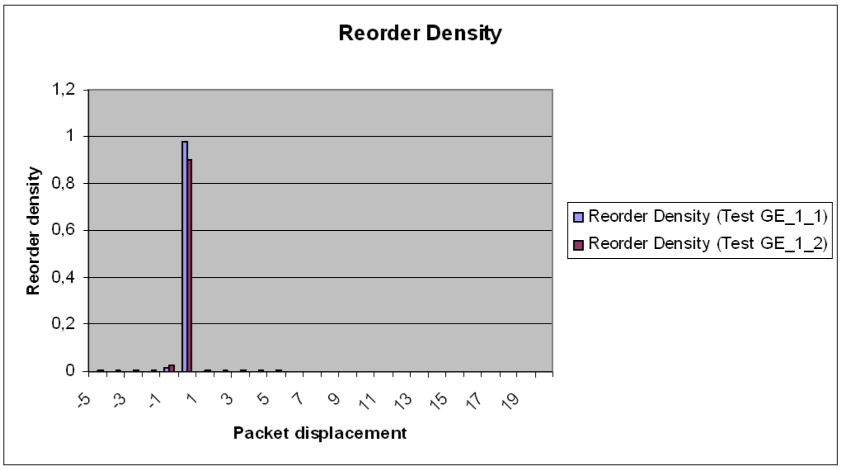


Figure 21: Reorder density distribution for scenarios GE_1_1 and GE_1_2

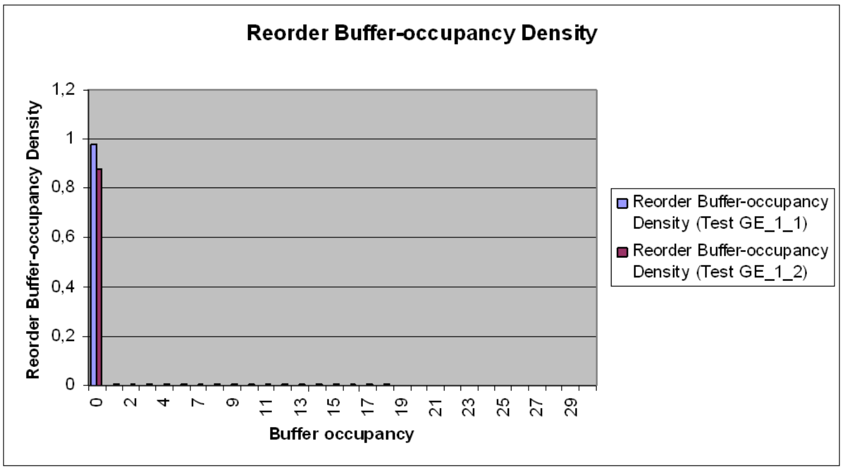


Figure 22: Reorder buffer-occupancy density distribution for scenarios GE_1_1 and GE_1_2

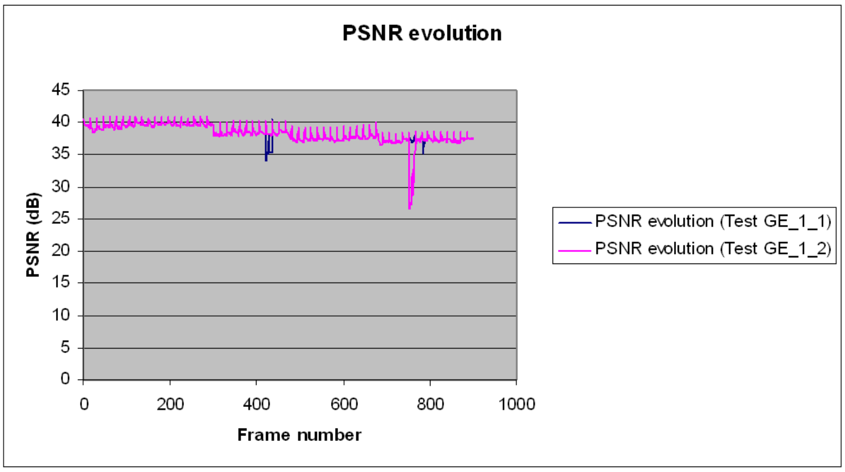


Figure 23: PSNR evolution for scenarios GE_1_1 and GE_1_2

2.4.5 Conclusions

In terms of delay and QoS, the Adaptive HARQ-based solution appears as a great trade-off compared to traditional reordering solutions in reception. The PSNR emerging from this technique is not so degraded whereas the average delay of reception is clearly favored. To privilege delay, techniques consisting in reducing the transmissions of images with a NRI value equal to zero could be developed. Several solutions are possible : transmitting images with a NRI_0 value at the end of the transmission, reducing the persistence value used for retransmissions for these particular images, delaying them gradually... The second option was tested in the simulations presented above. Maybe, it would be interesting to go further. Thus, linking these adaptations to scheduling policies at the queuing layer, new solutions against congestion issues could be investigated.

3 Multi Camera System

3.1 Introduction and State-of-the-art

In recent years the study of visual sensor networks (VSNs) has become a very attractive field for the scientific research. VSNs are characterized by huge capabilities and they support a great number of innovative applications. In particular the VSNs are employed for remote monitoring applications where direct visual information are essential for controlling and managing particular events [76]. VSNs are mainly formed by visual sensor nodes more commonly called video camera nodes. Each node integrates the image sensor and some embedded processing units. In such applications where cameras have fixed and easy-to-reach positions they can be connected via cable; in other cases they are equipped with a wireless radio interface.

VSNs are a useful tool in a number of social, research, educational applications, including:

- *surveillance*: protection of large area (airports, subways, stadiums, etc.) where detection and tracking mechanisms are required to localize possible intruders. This is one of the primary applications of VSNs, with a great number of cameras disseminated around the area of interest [29].
- *environmental monitoring*: video-cameras are deployed to monitor inaccessible or wild areas with the aim to acquire environmental images of the scene. Typical examples of such applications are: disaster sites management, traffic control in highways and natural environments monitoring such as forest and deserts.
- *telepresence and remote visiting systems*: the usage of multiple cameras permits a remote user to visit public buildings such as museums and art galleries and to participate to video conferences. Sometimes the underlying camera networks are developed to provide a user with visual information with arbitrary viewpoint [53]. In this way the user can explore the scenario of interest "moving" inside of it and changing the angle of visualization.
- *teleconsultation and distance healthcare*: in this case the VSNs constitute a useful instrument to aid remote medical assistance and to support clinical therapy from a distance. This is especially important for people with limited access to the healthcare system: people with disabilities, individuals living in remote rural areas or emergency scenarios or involved in accidents, with multiple injured people.

3.1.1 Video transmissions in multi-camera systems

One of the most critical issues concerning the VSNs consists in the large amount of data that each camera collects from the monitored environment. Differently from typical sensors (i.e. temperature or pressure sensors that provide measurements of unidimensional physical data), each image sensor collects a bi-dimensional set of data representing a captured image. The nature of the collected data entails complex tasks of information analysis and processing, as well as huge amount of data to be transmitted through the network. Moreover the majority of applications require real-time transmissions and this means strict requirements concerning the maximum allowable delay. For example the problem of the channel bandwidth optimization in the context of VSNs has been addressed in [83]. In that work a minimum bandwidth, for each selected camera, was guaranteed. Furthermore the remaining available resources were dynamically allocated considering the camera selection policy and the video contents.

3.1.2 Embedded processing

Based on the previous considerations, the optimal trade-off between the information accuracy and the network efficiency is to be pursued. To this purpose the embedded camera processing assumes an important role. The objective of the local processing consists in reducing the data flows to transmit or in providing some information about the context in order to assist the selection of proper acquisition and transmission policies. The local processing can be performed with different levels of intelligence, depending on the context on which the camera system is deployed. The most common processing operations include: background subtraction for motion/object detection [29],[21], edge detection and more complex algorithms such as feature extraction, object classification and scene reasoning [81],[56].

3.1.3 Sensor Collaboration

In some specific applications, such as detection and tracking of targets,VSNs need to cover wide areas and require highly detailed information. For this reason, the camera network manages a great number of information flows derived from a redundant number of interconnected nodes. The overall quality information can be increased by investigating the correlations between the individual camera data flows. A typical example is when two or more cameras have partially or completely overlapped FoVs. In this case the collaboration between sensors is defined as *spatial-based*. Many advantages can be obtained by spatial collaboration: for instance, through cross-validation, the reliability of the detection process is incremented. Furthermore some triangulation techniques can be applied permitting an accurate description of the target object (in terms of position and velocity) [59]. Even if two cameras are oriented in different directions, they can still cooperate. For example, in [34] a distributed scheme for a target tracking system has been studied. Given the position and the orientation of the nodes, a camera subset is dynamically selected for the object detection, while the remaining neighbouring cameras are used to extract and share the target features, which are correlated across the nodes.

3.1.4 Sensor management and camera selection

In the context of large camera networks, composed by a redundant number of camera nodes, the camera selection and management play a fundamental role. Sensor management defines the policies for scheduling the camera nodes activity. In each moment a subset of cameras has to be selected in order to perform continuous monitoring and collect information with the required quality. The purpose is to provide visual information able to satisfy the specified application requirements. Several cost metrics for the camera selection are investigated in the literature. In particular, the camera selection is of primary importance if the network is composed by self-supplied wireless camera nodes. In this case each camera is managed with the aim of defining the proper task and the activity period. The selected subset of involved cameras can be changed over time in order to balance the camera energy consumption. In [84] the authors propose a method to manage the power consumption of camera nodes. In the proposed solution each node is awake for a certain period of time and the camera node enters a low-power configuration based on the time-out status of its neighbouring cameras. Again, geometrical considerations about the placement and the orientation of the nodes are taken into account to defined further camera selection cost metrics. Each camera node is characterized by a directional sensing model: video cameras capture objects/scene images placed in different spatial positions. The gathered information are dependent on the direction on which the camera is oriented and the 3D viewing volume defined by the camera FoV. Several works in the literature analyze these aspects. In [75] a camera selection algorithm is provided with the aim to reconstruct a view from a user-specified view point. In particular two cost metrics are defined: the coverage-aware solution considers the remaining energy of the nodes and the overall coverage of the indoor space, while the quality-aware favours the selection of cameras with the most similar angle of view with respect to the user viewpoint. The video quality information also depends on the placement of the cameras. The optimal placement of the camera nodes in use is also addressed in recent works. An example is reported in [21], in which a good camera-network deployment is proposed mainly based on the target position. In its turn, [82] provides a model to manage a set of cameras that is basically focused on the occupancy of the monitored area. Only a limited number of cameras are involved and they send a limited amount of data in accordance to the energy-saving requirements and the bandwidth constraints. the final goal consists of minimizing the area potentially occupied by the moving objects and finding the number of cameras necessary to provide a sufficient visual hull area for each object.

3.2 A preliminary camera selection technique for healthcare and safety applications in emergency areas

In the context of healthcare and safety, multiple camera systems can support a great number of novel video-based services. In particular, some challenging applications that can be performed thanks to VSNs include real-time consulting and diagnosis from a remote location, transmission of multimedia contents from emergency scenarios and remote surgical assistance. The potential benefits of using camera network systems are manifold and include:

1. the possibility to visit more patients in a limited amount of time,
2. the possibility to have an overall control of the involved injured people and to coordinate the work of the first-aid personnel in emergency areas,
3. the opportunity to select the most suitable viewpoint in tele-diagnosis and remote medical assistance.

Moreover a wireless camera network, developed for this kind of scenarios, has to satisfy most of the requirements and constraints mentioned in the previous paragraph. For example a reasonable placement of the nodes has to be guaranteed in order to monitor the whole area of interest. Furthermore a redundant deployment of the nodes assures the advantages of spatial collaboration and provides useful information even if some cameras are not available or occluded. In addition, the camera selection policy has to be designed with the aim to collect information with a high level of accuracy. Only few works in this research field specifically address the image/video quality perceived by the final user as a main criterion for the camera selection problem. On the contrary we remark that in medical applications the QoE for the remote user is of crucial importance. For example, in [62] the authors choose the camera subset that best provides the quality of the reconstructed camera view. The basic idea proposed in [62] is to divide the 3D viewing volume (frustum) of each camera in smaller unit volume (sub-frustum). Then the selection criterion is based on the fact that the farther an object is from the centre of a sub-frustum the lower will be the quality of its representation in the captured images. Thus, based on the distance of each point from the sub-frustum centre, they select the camera that can display this point with the minimum visualization error. In [62] the concept of quality is strictly correlated to the position of the object of interest.

3.2.1 Assumptions

In this preliminary work [15] several assumptions have been made to facilitate the description of the camera network systems. We have used the "pinhole camera" model to describe the camera-nodes. The pinhole camera is the most used ideal model, for mapping a 3D scene into a 2D image. In Fig. 24.a the pinhole camera model has been depicted. The ray of lights coming from the different points in space pass through the optical centre *C* and the specular image of the scene is formed on the image plane *I*. Each point *M* in the 3D space corresponds to a point *m* projected onto the 2D image. To model our camera nodes more realistically we have assumed that each camera has a specific FoV, which delimits the portion of the 3D area that can be monitored. Only the points lying inside the FoV of the camera can be visualized in the image. Thus we have enriched the simple pinhole model considering the FoV. Each camera is characterized by horizontal and vertical FoVs parameters (one per image dimension). In Fig. 24.b the horizontal FoV is depicted. It can be notice that the FoV depends on the image dimension and on the *focal length f* defined as the distance between the focal plane (where the optical centre is placed) and the image plane.

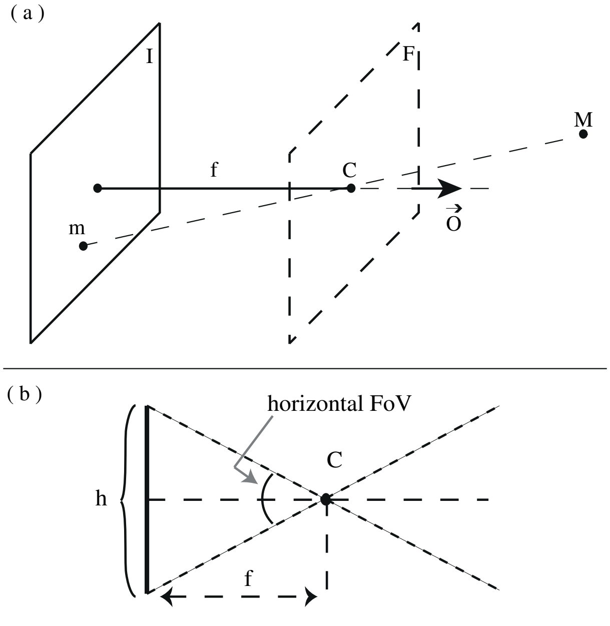


Figure 24: The "pinhole" camera model (a). The horizontal FoV (b).

We have also assumed that the position and the orientation of the cameras are known.

3.2.2 Preliminary definition of a ranking criteria for camera selection

In this section we propose a novel camera selection algorithm [15], which selects one or more cameras taking into account a specific ranking criteria mainly related to the quality of the visual representation of the object of interest. The proposed context-aware approach has been developed with the objective to:

- maximize the QoE for the final user,
- optimize the usage of distributed hardware resources,
- optimize the usage of radio resources for video transmissions.

Suppose we have a certain area to be monitored through the usage of a multi camera system; different wireless cameras with monitoring functionalities are placed in different positions.

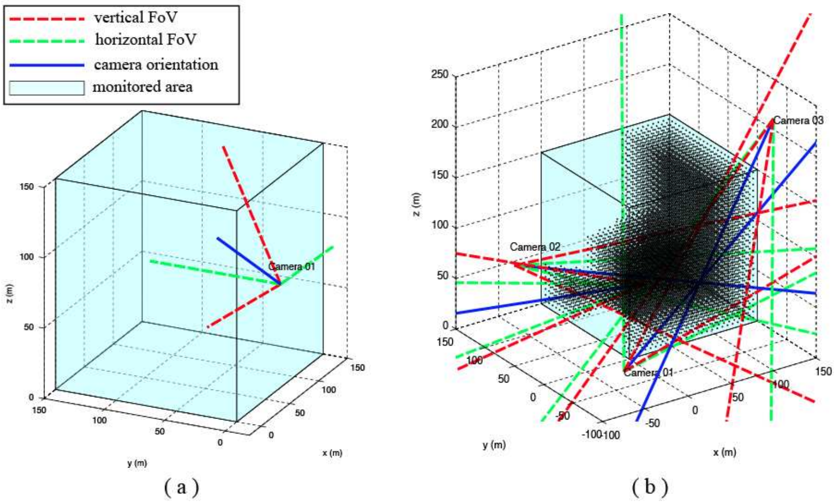


Figure 25: Camera system composed by a single node. In evidence the FoVs (a). Camera system composed by 3 nodes. The portion of the area monitored by the system is depicted in black (b).

In Fig. 25.a, a camera system composed by a single node has been depicted. In Fig. 25.b a generic system with 3 cameras is illustrated. Each camera covers a particular portion of the considered area depending on its orientation and FoV. In the case of two or more overlapped FoVs some portions of the area are commonly covered by multiple cameras (Fig. 25.b). The selection algorithm determines the best camera candidate to acquire visual information about a certain point in the monitored area based on specific raking criteria. The ranking rules we have considered in this preliminary phase are based on:

- the position of each camera,
- the orientation of each camera,
- the resolution of each camera.

For the sake of clarity, we show through an example, how the camera selection, based on these heterogeneous information, can optimize the final image quality of a target object. In Fig. 26 a couple of cameras capture an object placed in different positions. The goal consists in determining which camera provides the image of the object with the best resolution. We have also reported the image planes according to the different resolution of the cameras (a more dense grid corresponds to a higher resolution). In Fig. 26.a the cameras have the same resolution and the object is placed at a different distance from them. On the contrary in Fig. 26.b the cameras are characterized by different resolutions and the object is placed at the same distance from them. Increasing the distance (see camera C2 in Fig. 26.a), the projection of the object on the image plane gets smaller dimensions. In other words the closer is the object to the camera, the bigger and more detailed will be its representation on the image plane. Based on these considerations, the final image quality of the camera C1 is higher than that of camera C2. Increasing the camera resolution (see camera C2 in Fig. 26.b) high-level of details are still guaranteed even if the object is placed far away from the camera.

Changing the position of the object and the camera resolution entail an object description on the image plane characterized by a different number of pixels. In the two example of Fig. 26, the object is described with a different

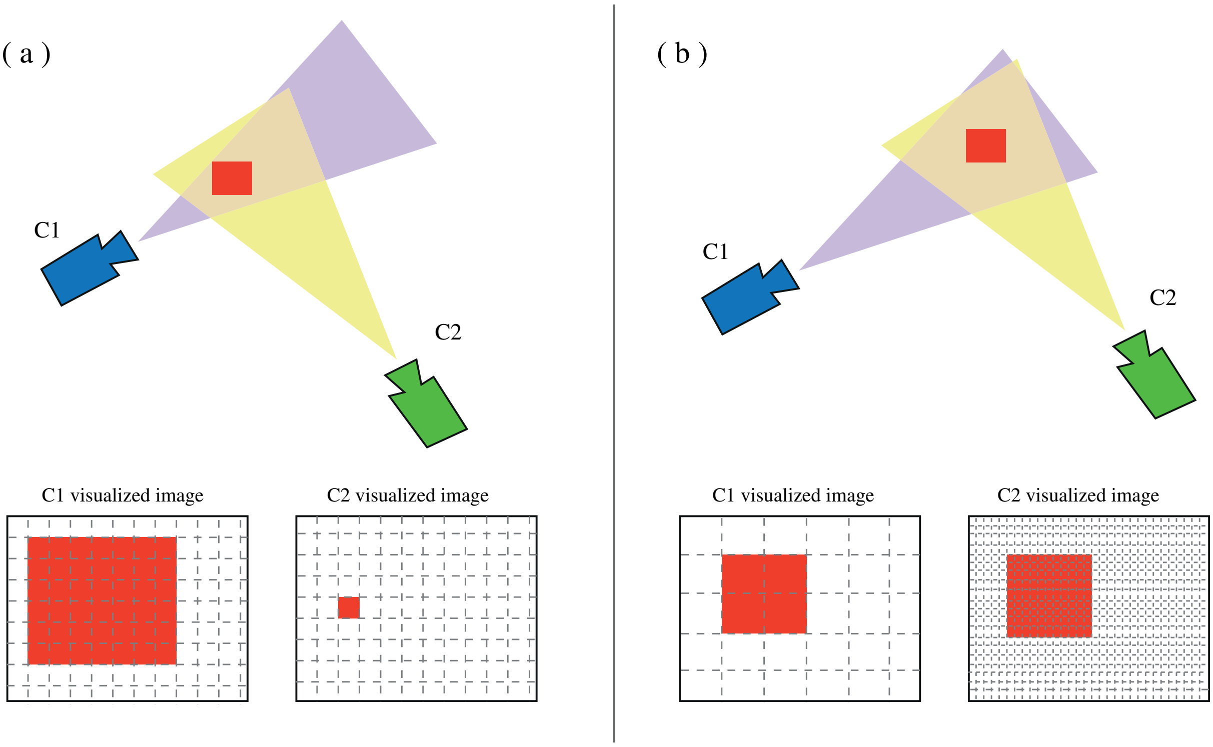


Figure 26: Two cameras are monitoring the same object. In (a) cameras have the same resolution and the object is placed at different distances from them. In (b) the object is at same distance from the cameras but they have different resolution.

number of pixels per object section area. The best image quality is obtained by the camera that guarantees the best trade-off in terms of distance and resolution.

Considering the scenario in Fig. 25.b we have discretized the monitored area in small sub-area unit; for each of them the proposed algorithm selects the camera that describes the given sub-area with the highest number of pixel per area. For the sake of clarity we graphically show the camera selection results for a set of sub-areas lying on the same plane perpendicular to the z-axis. In Fig. 27 and in Fig. 28 we report the representation of the considered plane; each squared element corresponds to a different sub-area. For each sub-area we compute the pixel per object section area; lighter colors correspond to high values of pixels values per squared meter, on the contrary when the number of pixel is decreasing the color tends to the dark. In Fig. 27 we have considered a camera system composed by a single node (Fig. 25.a). In Fig. 28 we have considered a camera system with 3 nodes (Fig. 25.b): for each sub-area the selected value of pixel per unit area is the highest possible among the set of cameras. The corresponding camera selection output is reported in Fig. 29, different sub-area colors correspond to different cameras. To capture the image of an object placed in the monitored area, the camera sensor capable to provide the object description with the highest resolution in terms of pixel per object unit area is selected.

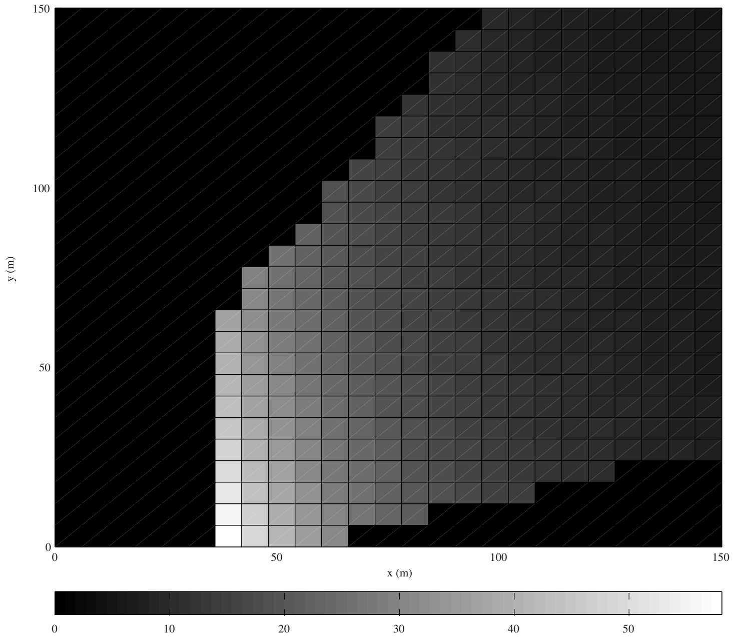


Figure 27: Pixel per unit area computation considering the scenario depicted in Fig. 25.a.

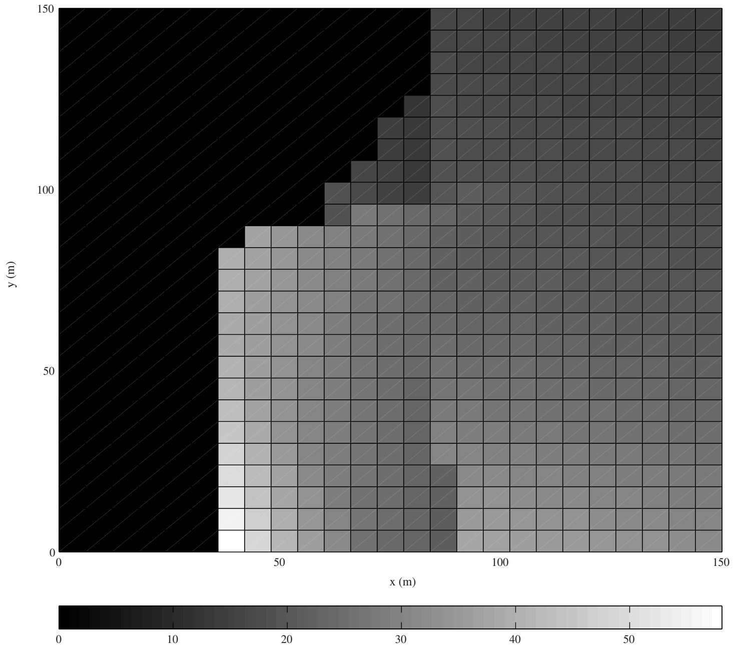


Figure 28: Pixel per unit area computation considering the scenario depicted in Fig. 25.b.

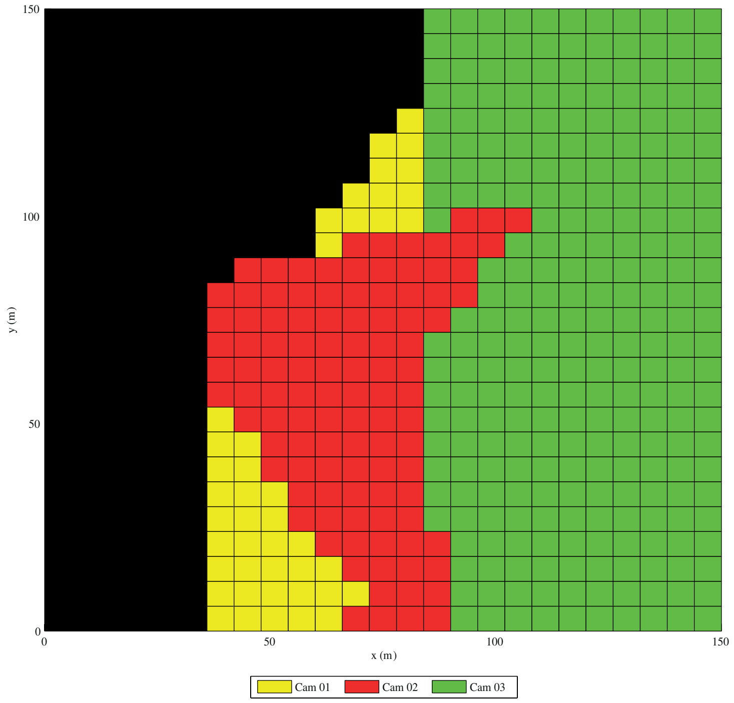


Figure 29: Camera selection results considering the scenario depicted in Fig. 25.b.

4 Packet-Level correction codes

In this section, we detail the progresses and achievements in the framework of packet-level codes to be used in the CONCERTO project. In packet-level coding, redundant packets (also called repair packets) are generated at the transmitter side, starting from a set of data packets (also called source packets) incoming from the upper layers. This redundancy is aimed at enabling the recovery of packet erasures and, possibly, the correction of bit errors at the receiver side. In video transmissions, packet-level codes are usually implemented at application or transport layer, with RTP or UDP/UDP-Lite protocols.

Packet-level coding solutions entail several important benefits in healthcare, safety and emergency scenarios. First of all, the adoption of powerful data protection techniques is of fundamental importance when the communication process involves information about people's health, e.g. for remote diagnosis purposes. In the second place, when heterogenous data streams (such as videos from mobile cameras, electrocardiograms, F.A.S.T. information, etc.) are multiplexed and transmitted to a remote user, prioritization and content-adaptive protection strategies can be performed through packet-level coding schemes operating at the application or transport layers. The advantages offered by such coding schemes become even more evident in the case of broadcast/multicast multimedia communications, thanks to their capacity to recover lost information without requiring retransmissions. Finally, in the context of telemedicine applications, a low-latency in the information delivery becomes often a primary requirement. Interactive communications, in which a remote operator is able to "navigate" through complex multimedia contents (e.g. zooming, pausing/rewinding, changing viewpoint, etc.), usually require very low transmission delay, often incompatible with traditional automatic retransmission request (ARQ) solutions.

In the context of packet-level coding, two main directions of investigation have been followed during the inital phase of the CONCERTO project. The first one is relevant to packet-level codes constructed on Galois fileds of order larger than 2 for erasure recovery, under efficient maximum likelihood (ML) decoding. In this framework, tight upper and lower bounds have been developed with the purpose to evaluate both the additional coding gain achievable by constructing packet-level codes on finite fields $\mathbb{F}_q$ with $q > 2$, and the performance achievable by non-binary codes based on *sparse* parity-check matrices. This preliminary analysis is fundamental to characterize the ultimate performance achievable with non-binary packet-level codes and, as a consequence, to evaluate the possible advantages offered by this kind of solutions in the context of multi-source and holographic tele-medicine (MSHTM) applications. The second one concerns packet-level codes for the simultaneous correction of errors and erasures. The limit case where only correct packets or packets corrupted by undetected errors are delivered to the layer of the protocol stack has been considered so far. This may be thought as a worst case in which the packet validation system on the receiver side always fails (or is not present at all) so that no erasures are generated. A simple enhanced decoder for LDPC codes is proposed for this scenario, and preliminary numerical results for regular LDPC codes are presented. The capacity to correct bit-errors possibly present in the received stream can improve the multimedia communication quality experienced by the remote medical staff, allowing quick and precise tele-consultation services.

4.1 Packet Erasure Correcting Codes based on High-Order Galois Fields

Tight bounds on the block error probability of linear block codes over order-q finite fields for the q-ary erasure channel, under ML decoding, are developed. Upper and lower bounds are obtained for uniform parity-check ensembles and sparse parity-check ensembles. Exploiting the derived bounds, it is shown how already for short blocks and small $q > 2$ sparse ensembles attain block error probabilities close to those of idealized maximum distance separable (MDS) codes, down to low error probabilities, whereas in the same regime binary codes show visible losses with respect to the Singleton bound. Thanks to the accurate performance estimates, the developed bounds can support the design of near-optimum erasure correcting codes to be adopted in the CONCERTO scenarios.

4.1.1 Introduction and State-of-the-Art

The binary erasure channel (BEC) [20] is often used as a simplified model for data losses in wireless/wired digital networks. Thus, the design of powerful coding schemes under low-complexity decoding for the BEC has been a rich research area in the past years [45, 47, 14, 43, 72]. In this framework, a class of codes that gained a particular attention is the one of LDPC codes [26], which approach the erasure channel capacity under belief propagation (BP) decoding for very large block lengths [45, 47, 63, 37]. In [65, 13, 61, 66, 17, 67, 60], low-complexity ML erasure decoders for LDPC codes were proposed that allow achieving, with judiciously designed LDPC codes, performances close to those of idealized MDS codes.

Bounds on the performance (in terms, e.g., of block error probability) of random and sparse graph-based ensembles over erasure channels have been the subject of past and recent research. For example, the block error probability over the BEC for random (linear) binary ensembles was analyzed in [20, 11, 50]. In [28] the performance limits over the block erasure channel (BLEC) have been investigated. The error probability of LDPC code ensembles in the finite length regime under both BP and ML decoding was analyzed in [18] over the BEC. Moreover, tight upper and lower bounds on the error probability for some classes of Luby transform (LT) codes were established in [68].

The memoryless q-EC represents an extension of the BEC. In a q-EC, the channel input alphabet $\mathcal{X}$ has cardinality $|\mathcal{X}| = q$ and each symbol is either correctly received or erased. The q-EC may serve as a reference model for data losses in digital networks, as its binary counterpart [48, 40, 35]. It has also been adopted to model optical communication links under q-ary pulse position modulation (PPM) and photon counting receivers in absence of background radiation [51, 52], [80, Chapter 4]. Furthermore, it gives an alternative representation of a binary BLEC [28] where blocks of $\log_2 q$ bits are either correctly received or erased. In [23] the error exponents of random and MDS codes over the q-EC were derived and compared.

In this section of the deliverable, we develop some tight upper and bounds on the block error probability of linear block codes over the q-EC under ML decoding, assuming that the field order of the code matches the channel input alphabet size q. More in detail, we derive upper and lower bounds for dense (i.e., uniform) parity-check ensembles and sparse parity-check ensembles. Exploiting the developed bounds, we are able to show that sparse ensembles attain block error probabilities close to those of idealized MDS codes, also for short blocks and relatively small $q > 2$, down to low error probabilities, whereas binary codes suffer for visible losses with respect to the Singleton bound.

4.1.2 Adopted Notation

Definitions The memory-less BEC is characterized by a binary input alphabet $\mathcal{X} = \{0, 1\}$ and a ternary output alphabet $\mathcal{Y} = \{0, 1, E\}$. The channel transition probabilities are $p_{Y|X}(0|0) = p_{Y|X}(1|1) = 1 - \epsilon$ and $p_{Y|X}(E|0) = p_{Y|X}(E|1) = \epsilon$, where E represents an erasure and ϵ is the channel erasure probability. In a q-EC, the input alphabet $\mathcal{X}$ has cardinality q. We assume that $q = 2^l$ and that the input alphabet symbols are the elements of $\mathbb{F}_q$, the finite field of order q. Thus, $\mathcal{X} = \{0, 1, \alpha^1, \ldots \alpha^{q-2}\}$ where α is a primitive element of $\mathbb{F}_q$. The output alphabet is $\mathcal{Y} = \{0, 1, \alpha^1, \ldots \alpha^{q-2}, E\}$, and its cardinality is $q + 1$. The channel transition probabilities are $p_{Y|X}(x|x) = 1 - \epsilon$ and $p_{Y|X}(E|x) = \epsilon$, $\forall x \in \mathcal{X}$ (Figure 30).

In the following we denote by $w_{\mathsf{H}}(\mathbf{v})$ the Hamming weight of a vector $\mathbf{v}$ with elements in $\mathbb{F}_q$ (i.e., the number of its nonzero elements) and by $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_i v_i w_i$ the inner product between $\mathbf{v}$ and $\mathbf{w}$, where all operations are in $\mathbb{F}_q$.

The finite field order of the linear block code G used to communicate over the q-EC is assumed to match the channel input alphabet size. We define the *support of a codeword* $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) \in \mathsf{G}$ as the set of its nonzero coordinates. Formally, $\mathrm{supp}(\mathbf{x}) = \{j \in \{0, 1, \ldots, n-1\} \text{ s.t. } x_j \neq 0\}$. We also define the the *support set* of $\mathbf{x} \in \mathsf{G}$ as $\mathcal{S}_{\mathbf{x}} = \{\mathbf{x}' \in \mathsf{G} \text{ s.t. } \mathrm{supp}(\mathbf{x}') = \mathrm{supp}(\mathbf{x})\}$. This is the set of codewords in G having the same support as $\mathbf{x}$. (Note that $\mathbf{x} \in \mathcal{S}_{\mathbf{x}}$ by default.) Moreover, we define the *weight of a support* as the Hamming weight of the corresponding codewords.

When a codeword of G is transmitted over the q-EC, the resulting erasure pattern $\mathscr{I}_E = \{i_1, i_2, \ldots, i_e\} \subseteq \{0, 1, \ldots, n-1\}$, of generic size e, identifies an $m \times e$ sub-matrix $\mathbf{H}_{\bar{K}}$ of $\mathbf{H}$ composed of the e columns of $\mathbf{H}$ with indexes in $\mathscr{I}_E$. ML decoding fails if and only if $\mathrm{rank}(\mathbf{H}_{\bar{K}}) < e$ or, equivalently, $\mathrm{supp}(\mathbf{x}) \subseteq \mathscr{I}_E$ for some nonzero codeword $\mathbf{x}$. The corresponding block error probability over a q-EC with erasure probability ϵ is denoted by $P_B(\mathsf{G}, \epsilon)$. We further denote by $P_f(\mathsf{G}, e)$ the average decoding failure probability given that there are exactly e erased symbols in the received word and assuming as equiprobable all erasure patterns of cardinality e.

We consider next *parity-check ensembles*, i.e., ensembles of linear block codes induced by random parity-check matrices. More specifically, the parity-check ensemble induced by a random matrix $\mathbf{H}$ with entries in $\mathbb{F}_q$ is composed of the linear block codes over the same finite field representing the null spaces of all possible realizations of $\mathbf{H}$. The probability of each code is the sum of the probabilities of the corresponding realizations of $\mathbf{H}$.

In particular, we denote by $\mathcal{C}(n, m, q)$ the ensemble of linear block codes induced by an $m \times n$ parity-check matrix $\mathbf{H}$ whose entries are i.i.d. random variables $X \in \mathbb{F}_q$ with uniform p.m.f., i.e., $\Pr\{X = \beta\} = 1/q \, \forall \, \beta \in \mathbb{F}_q$.

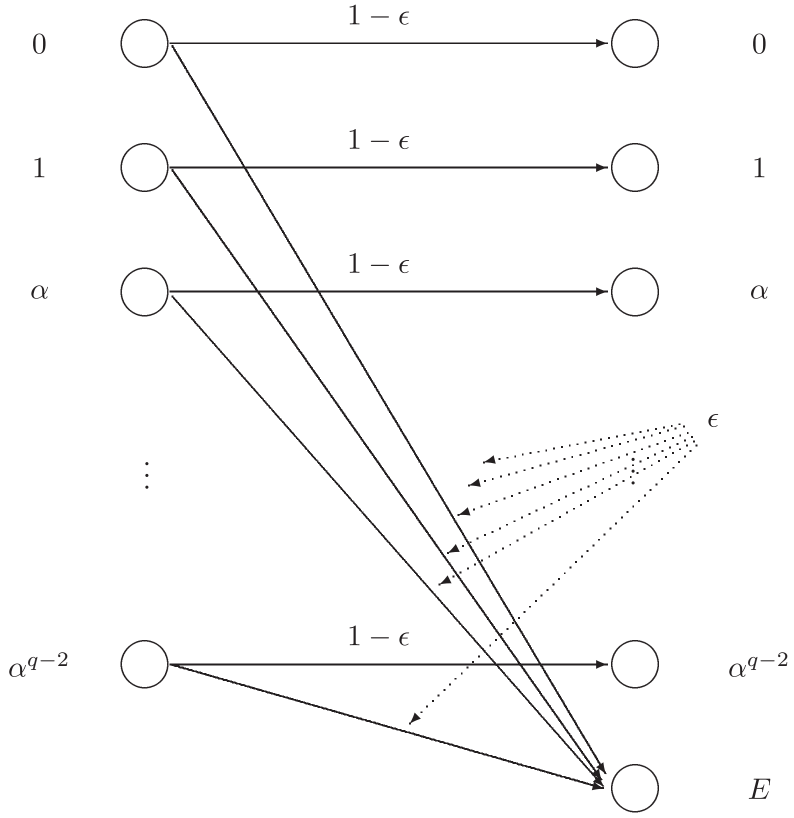


Figure 30: The q-ary erasure channel model.

We call this ensemble the *uniform parity-check ensemble*. We also denote by $\mathcal{S}(n,m,q,p)$ the ensemble induced by an $m \times n$ parity-check matrix $\mathbf{H}$ whose entries are i.i.d. random variables $X \in \mathbb{F}_q$ such that $\Pr\{X=0\}=1-p$ and $\Pr\{X=\beta\}=p/(q-1)\ \forall\beta\in\mathbb{F}_q\setminus\{0\}$. For small p sparse $\mathbf{H}$ matrices are characterized by a higher probability than dense ones, hence we call this ensemble the *sparse parity-check ensemble*.

Instead of considering a specific parity-check ensemble over $\mathbb{F}_q$, in some cases we will refer to a generic one, in the sense that the parity-check matrix $\mathbf{H}$ is characterized by a generic probability distribution. This ensemble will be denoted by $\mathcal{G}(n,m,q)$. The previous parity-check ensembles can be seen as particular instances of $\mathcal{G}(n,m,q)$.

Finally, we define as $\mathscr{C}(n,k,q)$ the ensemble of *all* codes with q^k codewords and length n, assuming a uniform probability distribution on such codes. Equivalently, $\mathscr{C}(n,k,q)$ is defined by a random code book of q^k codewords where each codeword is picked uniformly at random in $\mathbb{F}_q^n$. Note that the ensemble $\mathscr{C}(n,k,q)$ also includes *nonlinear* codes.

Singleton Bound The *Singleton bound* is defined as

$$P_B^{(\mathsf{S})}(n,m,\epsilon) \doteq \sum_{e=m+1}^{n} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e}. \tag{1}$$

The right-hand side of (1) is the probability that the number e of erased symbols exceeds the number m of equations (assumed linearly independent of each other) and then represents the block error probability of an idealized q-ary linear $(n,n-m)$ MDS code over the q-EC. Hence, it is a lower bound on the block error probability of every linear code of length n and rate $r \geq (n-m)/n$.

For any parity-check ensemble $\mathcal{G}(n,m,q)$, the average block error probability may be then always expressed as

$$\mathbb{E}_{\mathcal{G}(n,m,q)}[P_B(\mathsf{G},\epsilon)] = P_B^{(\mathsf{S})}(n,m,\epsilon) + \sum_{e=1}^{m} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} \mathbb{E}_{\mathcal{G}(n,m,q)}[P_f(\mathsf{G},e)]\,, \tag{2}$$

where $\mathbb{E}_{\mathcal{G}(n,m,q)}[\cdot]$ denotes expectation over the ensemble.

Berlekamp and Gallager Random Coding Bounds The performance of binary codes on the BEC was initially analyzed in [20], while the expected performance under ML decoding of a code drawn from $\mathcal{C}(n,m,2)$ was studied in detail in [11, 50]. More specifically, the average block error probability of codes in $\mathcal{C}(n,m,2)$ was upper bounded in [11] as (*Berlekamp random coding bound*)

$$\mathbb{E}_{\mathcal{C}(n,m,2)}\left[P_B(\mathsf{G},\epsilon)\right] < P_B^{(\mathsf{S})}(n,m,\epsilon) + \sum_{e=1}^{m} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} 2^{-(m-e)}. \tag{3}$$

In [18, App. B], the Berlekamp bound is re-derived as a union bound on the expected block error probability of the codes in $\mathcal{C}(n,m,2)$.

Since (3) provides an upper bound on the average block error probability for the parity-check ensemble $\mathcal{C}(n,m,2)$, which includes all the binary linear block codes of length n and rate $r \geq 1 - m/n$, it can be used an upper bound for the ensemble $\mathscr{C}(n,n-m,2)$ of binary block codes with length n and 2^{n-m} codewords, in the sense that there exists at least one code in $\mathscr{C}(n,n-m,2)$ whose block error probability (averaged over the code book) under ML decoding is upper bounded by the Berlekamp bound.

An alternative upper bound on the average block error probability for the ensemble $\mathscr{C}(n,n-m,2)$ is the Gallager random coding bound [27], recently extended to the ensemble $\mathscr{C}(n,n-m,q)$ [23]:

$$\mathbb{E}_{\mathscr{C}(n,k,q)}\left[P_B(\mathsf{G},\epsilon)\right] < e^{-nE_{q,\epsilon}(r)} \tag{4}$$

where $E_{q,\epsilon}(r)$ is the random coding exponent for the q-ary codes with rate $r = 1 - m/n$. On the q-EC, the exponent $E_{q,\epsilon}(r)$ is given by

$$E_{q,\epsilon}(r) = \begin{cases} -\ln\left(\frac{1-\epsilon}{q} + \epsilon\right) - r\ln q & 0 < \epsilon < \epsilon_c \\ \mathcal{D}(1-r,\epsilon) & \epsilon_c \leq \epsilon \leq 1-r \\ 0 & \epsilon > 1-r \end{cases} \tag{5}$$

where

$$\mathcal{D}(1-r,\epsilon) = (1-r)\ln\frac{1-r}{\epsilon} + r\ln\frac{r}{1-\epsilon}$$

is the Kullback-Leibler distance between two Bernoulli distributions with parameters ϵ and $1-r$, and

$$\epsilon_c = \frac{1-r}{r(q-1)+1}. \tag{6}$$

Both (3) and (4) (with $q = 2$) represent upper bounds on the block error probability of the best code in $\mathscr{C}(n,n-m,2)$. The bound (4) is known to be less tight than (3) down to low block error probabilities [19]. However, it becomes tighter at very low error probabilities.

Rank of Matrices with i.i.d. Uniform Elements over $\mathbb{F}_q$ For $e \leq m$, consider an $e \times m$ random matrix $\mathbf{A}$, whose elements are drawn independently at random with uniform distribution in $\mathbb{F}_q$. We define $P_e = \Pr\{\mathrm{rank}(\mathbf{A}) < e\}$ as the probability of $\mathbf{A}$ being not full-rank. We also denote by $\delta = m - e$ the number of rows in excess w.r.t. the number of columns. The probability P_e may be expressed as a function of e, δ, and q and is given by [38, 36]

$$P_e(e,\delta,q) = 1 - \prod_{i=1}^{e}\left(1 - \frac{q^{i-1}}{q^{e+\delta}}\right). \tag{7}$$

The following result was developed in [40].

Proposition 1. *The probability P_e can be bounded above and below as*

$$q^{-\delta-1} \leq P_e(e, \delta, q) < \frac{1}{q-1} q^{-\delta} \,. \tag{8}$$

Remarkably, the upper and the lower bounds in (8) are independent of the number e of columns. The two bounds converge for large q and the gap between them is very small for all q. The upper bound is very tight even for small values of e and q [40, 35].

4.1.3 Average Block Error Probability for Uniform Parity-Check Ensembles

We are now in a position to derive tight bounds on the expected block error probability under ML decoding of a linear block code G drawn from the uniform parity-check ensemble $\mathcal{C}(n, m, q)$, over a q-EC with erasure probability ϵ.

Theorem 1. *Let $\mathcal{C}(n, m, q)$ be the ensemble of linear block codes induced by an $m \times n$ random parity-check matrix $\mathbf{H}$ with i.i.d. entries X such that $\Pr\{X = \beta\} = 1/q \; \forall \beta \in \mathbb{F}_q$. Then the expected block error probability of a code G picked randomly in $\mathcal{C}(n, m, q)$, under ML decoding and over a q-EC with erasure probability ϵ, fulfills*

$$\mathbb{E}_{\mathcal{C}(n,m,q)} \left[P_B(\mathsf{G}, \epsilon) \right] \geq P_B^{(\mathsf{S})}(n, m, \epsilon) + \sum_{e=1}^{m} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} q^{-(m-e)-1} \tag{9}$$

and

$$\mathbb{E}_{\mathcal{C}(n,m,q)} \left[P_B(\mathsf{G}, \epsilon) \right] < P_B^{(\mathsf{S})}(n, m, \epsilon) + \frac{1}{q-1} \sum_{e=1}^{m} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} q^{-(m-e)} \,. \tag{10}$$

Proof. For a general parity-check ensemble $\mathcal{G}(n, m, q)$ the expected block error probability is given by (2). In the specific case of the $\mathcal{C}(n, m, q)$ ensemble, from Section 4.1.2 we have

$$\begin{aligned} \mathbb{E}_{\mathcal{C}(n,m,q)} \left[P_f(\mathsf{G}, e) \right] &= P_e(e, m-e, q) \\ &= 1 - \prod_{i=1}^{e} \left(1 - \frac{q^{i-1}}{q^m} \right) \,. \end{aligned} \tag{11}$$

Moreover, Proposition 1 yields

$$q^{-(m-e)-1} \leq \mathbb{E}_{\mathcal{C}(n,m,q)} \left[P_f(\mathsf{G}, e) \right] < \frac{1}{q-1} q^{-(m-e)}$$

from which the lower bound (9) and the upper bound (10) are obtained. □

Note that for $q = 2$, the upper bound reduces to the Berlekamp bound (3). As for the binary case, (10) holds for the ensemble $\mathscr{C}(n, n-m, q)$ of q-ary block codes with codeword length n and q^{n-m} codewords, in the sense that there exists at least one code in $\mathscr{C}(n, n-m, q)$ whose block error probability under ML decoding is upper bounded by the right-hand side of (10).

We are now interested in comparing the extension of Berlekamp bound in (10) with Gallager random coding bound (4). To this purpose, observe that ϵ_c in (6) is maximized for $q = 2$ and that it vanishes for large q. Hence, the error exponent is equal to $\mathcal{D}(1-r, \epsilon)$ for any $\epsilon \in [\epsilon^*/(r+1), 1-r]$ regardless of q. Moreover, the range of values of ϵ for which the error exponent depends on q vanishes as q increases. It follows that for moderate erasure probabilities, i.e., $\epsilon \geq \epsilon_c(q)$, Gallager random coding bound does not incorporate the effect of the channel input alphabet size, whereas the upper bound in (10) does, becoming tighter as q increases.

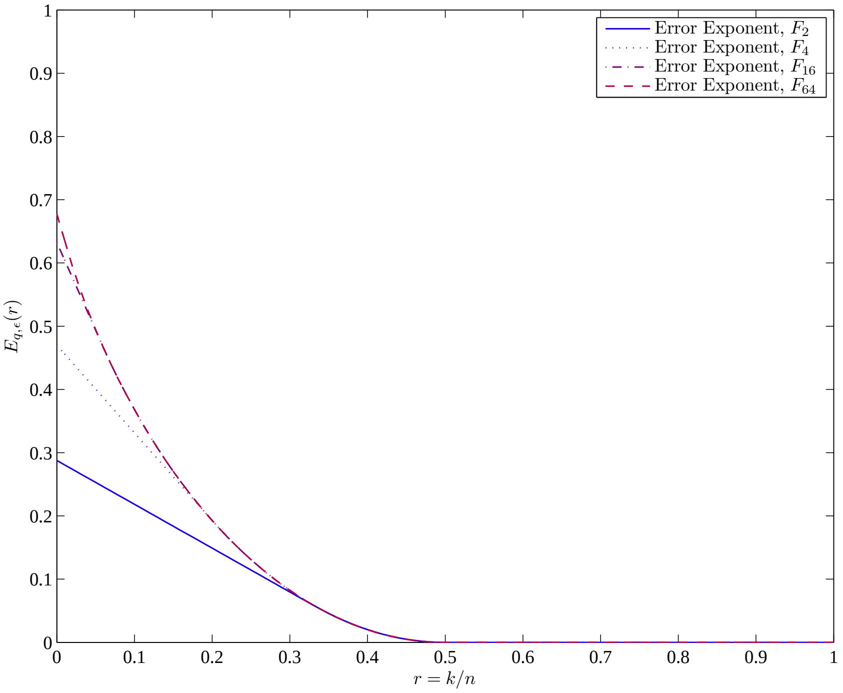


Figure 31: Plot of the error exponent $E_{q,\epsilon}(r)$ for $\epsilon = 0.5$ and $q = 2$, 4, 16, and 64.

Example 1. *Figure 31 depicts the exponent $E_{q,\epsilon}(r)$ for different field orders and $\epsilon = 0.5$. Observe that at rates close to the capacity $C = 1 - \epsilon = 0.5$ the exponent is independent of the field order. Figure 32 provides the lower and upper bounds* (9) *and* (10) *for $n = 120$ and $m = 60$, as well as the Singleton bound. They show to be tight for all q and get tighter for larger field orders. For $q = 16$ the upper bound* (10) *matches closely the Singleton bound down to low block error probabilities. Gallager random coding bound for the $\mathscr{C}(120, 60, q)$ ensemble with $q \to \infty$ is also depicted. As expected, the extension* (10) *of Berlekamp bound is tighter than Gallager bound for all $q > 2$, at least down to $P_B = 10^{-10}$.* [1] *Since at least one code in $\mathscr{C}(n, n-m, q)$ must exist whose performance is upper bounded by the q-ary extension* (10) *of the Berlekamp bound, at least one code in $\mathscr{C}(120, 60, q)$ exists, for any $q \geq 16$, approaching very closely the Singleton bound down to low block erasure probabilities. Moreover, due to tightness, extremely good codes (i.e., codes performing very close to the Singleton bound down to low block error probabilities) exist even in the* linear *(parity-check) ensemble.*

4.1.4 Average Block Error Probability for Sparse Parity-Check Ensembles

In this section, an upper and a lower bound on the expected block error probability over the q-EC for q-ary linear block codes drawn from $\mathcal{S}(n, m, q, p)$ are developed. The bounds are established by the following theorem.

Theorem 2. *Let $\mathcal{S}(n, m, q, p)$ be the ensemble of linear block codes induced by an $m \times n$ random parity-check matrix* $\mathbf{H}$ *with i.i.d. entries X such that $\Pr\{X = 0\} = 1 - p$ and $\Pr\{X = \beta\} = p/(q-1)$ $\forall \beta \in \mathbb{F}_q \setminus \{0\}$. Then the expected block error probability of a code* G *picked in $S(n, m, q, p)$, under ML decoding and over a q-EC with*

[1] Note that for the $\mathscr{C}(120, 60, 2)$ ensemble we have $\epsilon_c = 1/3$. Thus, down to $P_B \approx 10^{-5}$ the Gallager random coding bound for $q = 2$ coincides its extension to larger values of q.

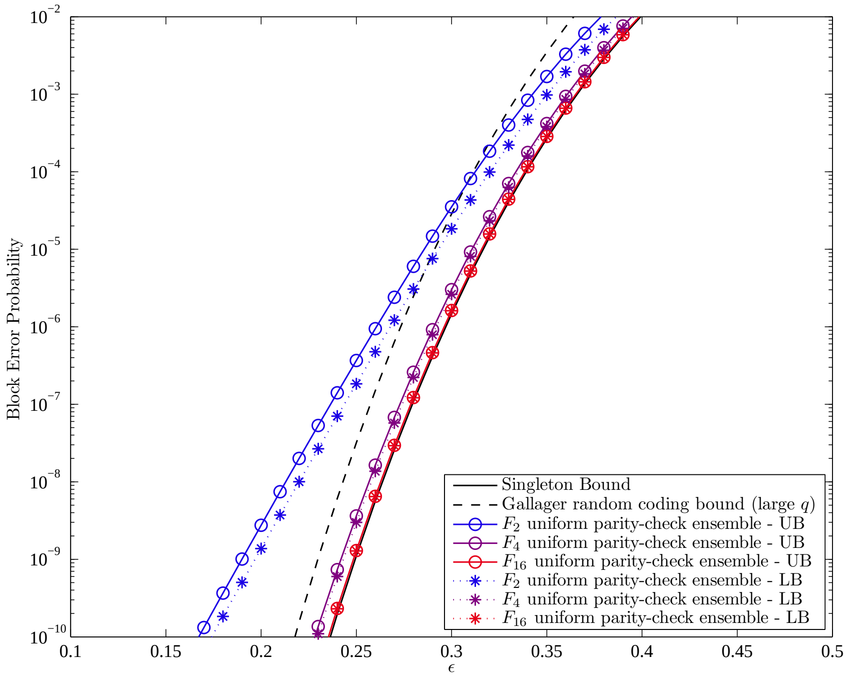


Figure 32: Lower and upper bounds (9) and (10) on $\mathbb{E}_{\mathcal{C}(120,60,q)}[P_B(\mathsf{G},\epsilon)]$, for the binary, 4-ary, and 16-ary erasure channel. The Gallager random coding bound (4) with the error exponent (5) for the ensemble $\mathscr{C}(120,60,q)$ is also depicted the case of large q. Note that the performance under ML decoding of at least one code in $\mathscr{C}(120,60,q)$ for $q \geq 16$ is upper bounded by the right-hand side of (10) with $q = 16$ and is lower bounded by the Singleton bound, thus achieving a near-optimum performance at least down to block error probability 10^{-10}.

erasure probability ϵ, may be upper bounded as

$$\mathbb{E}_{\mathcal{S}(n,m,q,p)}\left[P_B(\mathsf{G},\epsilon)\right] \leq P_B^{(\mathsf{S})}(n,m,\epsilon) + \sum_{e=1}^{m} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e}$$
$$\times \min\left\{1, \frac{1}{q-1} \sum_{t=1}^{e} \binom{e}{t} (q-1)^t \left(\frac{q-1}{q}\left(1-\frac{p\,q}{q-1}\right)^t + \frac{1}{q}\right)^m\right\} \qquad (12)$$

and lower bounded as

$$\mathbb{E}_{\mathcal{S}(n,m,q,p)}\left[P_B(\mathsf{G},\epsilon)\right] > \; P_B^{(\mathsf{S})}(n,m,\epsilon) + \sum_{e=1}^{m} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} \; \left[1-\left(1-(1-p)^m\right)^e\right] . \qquad (13)$$

Proof. We start by proving the upper bound. Consider an erasure pattern $\mathscr{I}_E$ of size $|\mathscr{I}_E| = e$ and its corresponding sub-matrix $\mathbf{H}_{\bar{K}}$. Via union bound we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}(n,m,q,p)}\left[P_f(\mathsf{G},e)\right] &= \Pr\left\{\operatorname{rank}(\mathbf{H}_{\bar{K}}) < e\right\} \\
&= \Pr\left\{\exists\, \mathbf{v} \in \mathbb{F}_q^e \setminus \{\mathbf{0}\} \text{ s.t. } \mathbf{H}_{\bar{K}} \mathbf{v}^T = \mathbf{0}^T\right\} \\
&\overset{\text{(a)}}{\leq} \min\left\{1, \frac{1}{q-1} \sum_{\mathbf{v}\in\mathbb{F}_q^e\setminus\{\mathbf{0}\}} \Pr\left\{\mathbf{H}_{\bar{K}} \mathbf{v}^T = \mathbf{0}^T\right\}\right\} \\
&= \min\left\{1, \frac{1}{q-1} \sum_{t=1}^{e} \binom{e}{t} (q-1)^t \Pr\left\{\mathbf{H}_{\bar{K}} \mathbf{v}^T = \mathbf{0}^T | w_{\mathsf{H}}(\mathbf{v}) = t\right\}\right\} \qquad (14)
\end{aligned}$$

where the $1/(q-1)$ factor in (a) is due to the fact that if some $\mathbf{v}_1 \in \mathbb{F}_q^e \setminus \{\mathbf{0}\}$ exists such that $\mathbf{H}_{\bar{K}}\mathbf{v}_1^T = \mathbf{0}^T$, then there certainly exist other $q-2$ vectors $\mathbf{v}_i \in \mathbb{F}_q^e \setminus \{\mathbf{0}\}$, $i = 2, \dots, q-1$, such that $\mathbf{H}_{\bar{K}}\mathbf{v}_i^T = \mathbf{0}^T$ and such that $\bigcup_{i=1}^{q-1}\{\text{event that } \mathbf{H}_{\bar{K}}\mathbf{v}_i^T = \mathbf{0}^T\} = \{\text{event that } \mathbf{H}_{\bar{K}}\mathbf{v}_1^T = \mathbf{0}^T\}$ (simply multiply $\mathbf{v}_1$ by the $q-2$ elements of $\mathbb{F}_q \setminus \{0, 1\}$). Note that there are exactly $\binom{e}{t}(q-1)^t$ vectors $\mathbf{v}$ of length e and Hamming weight t and that all of them are characterized by the same probability $\Pr\left\{\mathbf{H}_{\bar{K}}\mathbf{v}^T = \mathbf{0}^T | w_{\mathsf{H}}(\mathbf{v}) = t\right\}$.

Next, denoting by $\mathbf{h}$ the generic row of $\mathbf{H}_{\bar{K}}$ due to independence and identical distribution of the entries of $\mathbf{H}_{\bar{K}}$ we have

$$\begin{aligned}\Pr\left\{\mathbf{H}_{\bar{K}}\mathbf{v}^T = \mathbf{0}^T | w_{\mathsf{H}}(\mathbf{v}) = t\right\} &= \left(\Pr\left\{\langle \mathbf{h}, \mathbf{v}\rangle = 0 | w_{\mathsf{H}}(\mathbf{v}) = t\right\}\right)^m \\ &= \left(\Pr\left\{X_1 + X_2 + \cdots + X_t = 0\right\}\right)^m \end{aligned} \tag{15}$$

where X_1, X_2, ..., X_t are i.i.d. random variables in $\mathbb{F}_q$ each characterized by a p.m.f. p_X equal to that of the generic entry of $\mathbf{H}_{\bar{K}}$. The expression (15) may be evaluated using the Fourier transform over Abelian groups [79, Chapter 10], as follows. Let $\mathbb{G}_q$ be the additive group of $\mathbb{F}_q$ and χ be the generic character of $\mathbb{G}_q$, with the convention that χ_0 is the principal character of $\mathbb{G}_q$, i.e., $\chi_0(a) = 1 \in \mathbb{C}$ for all $a \in \mathbb{G}_q$ (as usual, $\mathbb{C}$ denotes the field of complex numbers). Due to independence, the p.m.f. p_Y of $Y = X_1 + X_2 + \cdots + X_t$ is given by the convolution of the pp.m.f. of the t summands. Interpreting pp.m.f. as functions from $\mathbb{G}_q$ to $\mathbb{C}$, the Fourier transform $\hat{p}_Y$ of p_Y is given by $\hat{p}_Y(\chi) = [\hat{p}_X(\chi)]^t$, where $\hat{p}_X$ is the Fourier transform of p_X. It is easy to show that[2]

$$\hat{p}(\chi) = \begin{cases} 1 & \text{if } \chi = \chi_0 \\ 1 - \frac{qp}{q-1} & \text{elsewhere} \end{cases}$$

from which, applying the inverse Fourier transform, we obtain

$$\Pr\{X_1 + X_2 + \cdots + X_t = 0\} = \frac{q-1}{q}\left(1 - \frac{pq}{q-1}\right)^t + \frac{1}{q}\,. \tag{16}$$

Substituting back (16) into (15) and then (15) into (14), from (2) we obtain (12).

The lower bound may be proved as follows. Considering again an erasure pattern $\mathscr{I}_E$ of size $|\mathscr{I}_E| = e$ and its corresponding sub-matrix $\mathbf{H}_{\bar{K}}$, the probability $\mathbb{E}_{\mathcal{S}(n,m,q,p)}\left[P_f(\mathsf{G}, e)\right] = \Pr\left\{\text{rank}(\mathbf{H}_{\bar{K}}) < e\right\}$ is lower bounded by the probability that $\mathbf{H}_{\bar{K}}$ has at least one all-zero column, this latter probability being $\sum_{t=1}^{e}\binom{e}{t}(1-p)^{mt}(1-(1-p)^m)^{e-t} = 1 - (1-(1-p)^m)^e$. The lower bound (13) is then obtained from (2). □

Interestingly, the same proof technique adopted to prove (12) may also be used to prove (10), assuming that the entries of $\mathbf{H}_{\bar{K}}$ are i.i.d. random variables with a uniform probability distribution over $\mathbb{F}_q$. Note also that the lower bound is independent of q.

Example 2. *Figure 33 shows the upper bounds* (12) *on the average block error probability for* $\mathcal{S}(120, 60, q, p)$ *sparse parity-check ensembles with* $q = 2$*,* 4*, and* 16*, and matrix densities* $p = 1/3$*,* $1/4$*, and* $1/5$*. On the same plot, the lower bounds* (13) *are provided. Observe that, while in the waterfall region the upper bounds tend to get closer to the Singleton bound for increasing field orders* q*, in the error floor region the upper bounds become almost independent of* q*, and are nearly undistinguishable from the corresponding lower bounds (which are independent of the field order). Thus, for sparse parity-check ensembles the average error floor performance is dominated by the performance of the codes in the ensemble having minimum distance* 1*, i.e., codes having codeword symbols that do not participate in any parity-check equation. The probability of drawing a code with minimum distance* 1 *drops remarkably by increasing the matrix density* p*. For instance, for* $p = 1/3$ *the error floor appears at block error probability about* 10^{-9}*, while for* $p = 1/5$ *the floor rises up at a block error probability about* 10^{-4}.

4.2 Packet-Level Codes in Error-and-Erasure Environments

We now consider packet-level LDPC codes in scenarios where packet errors as well as packet erasures may be present. As a preliminary analysis, we start by investigating the extreme case in which only correct packets or packets corrupted by undetected errors are delivered to the layer of the protocol stack in which the packet-level

[2] Since we consider fields of order $q = 2^l$, the Fourier transform reduces to the Hadamard transform [49].

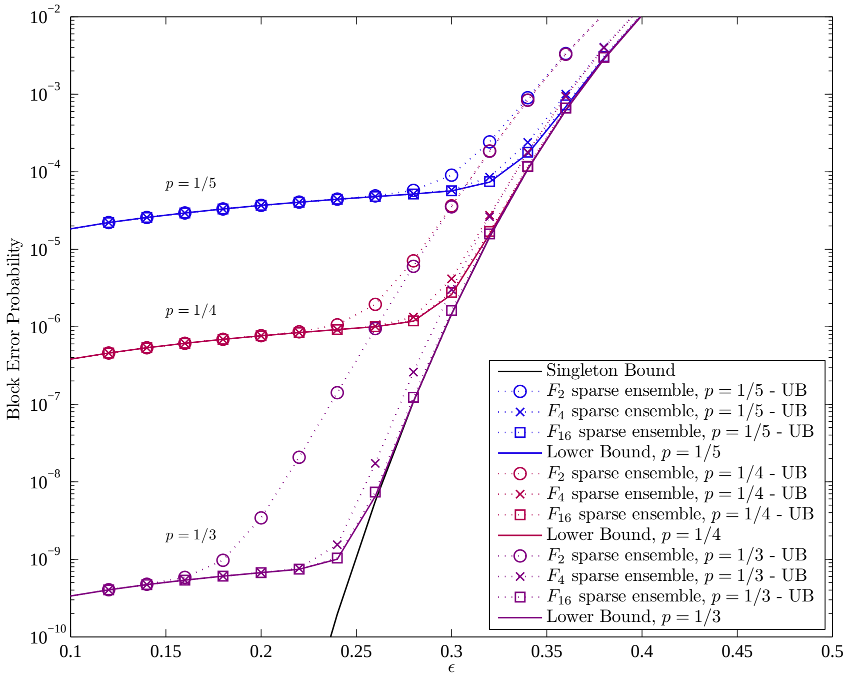


Figure 33: Upper and lower bounds (12) and (13) on the average block error probability for sparse parity-check ensembles with $n = 120$ and $m = 60$, for the binary, 4-ary, and 16-ary erasure channel. Matrix density p set to $1/5$, $1/4$ and $1/3$. Note that the lower bound is independent of the field order.

decoder operates. This is an extreme worst case in which the packet validation system on the receiver side fails with probability 1 (i.e., no errors are detected and then no erasures are generated). In a more realistic situation, that will be analyzed in subsequent CONCERTO deliverables, the decoder faces both undetected errors and erasures.

Next, we consider VBD, a simple yet powerful iterative decoding technique for codes operating on packets (or vector-symbols) and channels that introduce packet-wise errors. Examples are virtual channels after inner decoding in concatenated schemes or packet transmission channels where each single packet is not verified through a Cyclic Redundancy Check (CRC). We implement and analyze through simulation a simple improvement of the VBD algorithm, consisting of a Gaussian elimination (GE) based erasure decoding stage whenever VBD fails, i.e., when $\alpha > 0$ packets remain unverified. The modification allows the recovery of the α unverified packets with a complexity of $\mathcal{O}(\alpha^3)$ by making use of the inherent code structure. Monte Carlo simulations on LDPC codes show the benefit of the proposed algorithm.

4.2.1 Introduction and State-of-the-Art

In those applications where linear block codes are used for packet recovery, packets may be correctly received, discarded or marked as “bad" thus generating erasures (if they do not pass some validation test, e.g., based on a CRC), or incorrectly marked as “good" and delivered to the upper layers even if they are corrupted (undetected error). The error probability may either small in comparison with the erasure probability (if the error detection code keeps a moderate-low undetected error probability [41]) or even comparable (if not predominant, if the employed error detection code is too weak).

In the limiting case, when no error detection is employed or when it is assumed that it fails with probability 1, we obtain a q-ary symmetric channel (q-SC). The capacity of the q-SC with $q = 2^l$ and error probability ϵ is

$$C = l - h(\epsilon) - \epsilon \log_2 \left(2^l - 1\right) \tag{17}$$

where $h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function. Note that already for packet sizes l in the order of a few tens of bits q becomes large. It can be easily checked that for large q the capacity C in

(17) normalized to l approaches the capacity of an erasure channel with an erasure probability equal to the error probability ϵ,

$$\lim_{l\to\infty} \frac{C}{l} = 1 - \epsilon.$$

Algorithms that allow good error correction capabilities over the q-SC with large q were originally proposed in [53, 46]. The techniques introduced in [46] are commonly referred to as VBD algorithms and are applied to LDPC codes [26]. Further developments of the decoding on the q-SC were proposed in [69, 71, 70]. In this deliverable, we elaborate on the two simplest VBD algorithms of [46] by introducing a final re-processing stage of the remaining unverified packets, whenever VBD fails. We perform an erasure of the residual unverified packets followed by a recovery through maximum a-posteriori (MAP) erasure decoding. More specifically, we add a GE-based erasure decoding stage when $\alpha > 0$ packets remain unverified. The suggested modification allows the recovery of the α unverified packets with a complexity $\mathcal{O}(\alpha^3)$. A similar concept was proposed in [71, 70]. However, in [71, 70] the residual unverified symbols are recovered by an outer erasure correcting code, whereas we exploit the check equations left unused by the VBD algorithm to proceed with the final recovery stage.

4.2.2 Preliminary Definitions

We consider the application of a linear (n, k) block code to protect packets transmitted over a packet error channel. We restrict ourselves to binary linear block codes although the following derivations could be easily extended to the more general non-binary case. The binary linear block code imposes a set of $n - k$ binary parity constraints on the n codeword packets, also referred to as *symbols*. Symbols can be thus considered either as vectors of l bits, hence as elements of $\mathbb{F}_2^l$, or as elements of a finite field $\mathbb{F}_q$ of order $q = 2^l$.

The packet error channel is modeled as a q-SC with error probability ϵ. The (n, k) binary linear block code parity-check matrix is denoted by $\mathbf{H}$. It is a binary $m \times n$ matrix with $m \geq n - k$. The Tanner graph associated with $\mathbf{H}$ is denoted by $\mathcal{G}$, where $\mathcal{G} = (\mathscr{V}, \mathscr{C}, \mathscr{E})$ consists of a set of n variable nodes (VNs) $\mathscr{V}$, a set of m check nodes (CNs) $\mathscr{C}$, and a set of edges $\mathscr{E}$. Each edge $e_{i,j} \in \mathscr{E}$ connects a VN $V_j \in \mathscr{V}$ to a CN $C_i \in \mathscr{C}$ if and only if $h_{i,j} = 1$. The j-th VN represents the j-th codeword packet. We associate a state to each VN and to each CN. The operator $\mathrm{S}(\cdot)$ returns the node state. We have $\mathrm{S}(V_j) \in \{\mathsf{V}, \overline{\mathsf{V}}\}$, i.e., a VN can be either *verified* (V) or *unverified* ($\overline{\mathsf{V}}$). Similarly, we have $\mathrm{S}(C_i) \in \{\mathsf{C}, \overline{\mathsf{C}}\}$, i.e., a CN can be either *confirmed* (C) or *unconfirmed* ($\overline{\mathsf{C}}$).

4.2.3 Enhanced Verification-Based Decoding

In its most simple setting (Algorithm A, in the following), VBD works as follows [46]:

i. Initialization. At the beginning all VNs are marked as unverified ($\overline{\mathsf{V}}$) and all CNs as unconfirmed ($\overline{\mathsf{C}}$).

ii. Process in parallel all CNs by performing

  a. Verification. If all VNs connected to a CN fulfill the parity, they are set to verified (V) and the check node is set to confirmed (C).

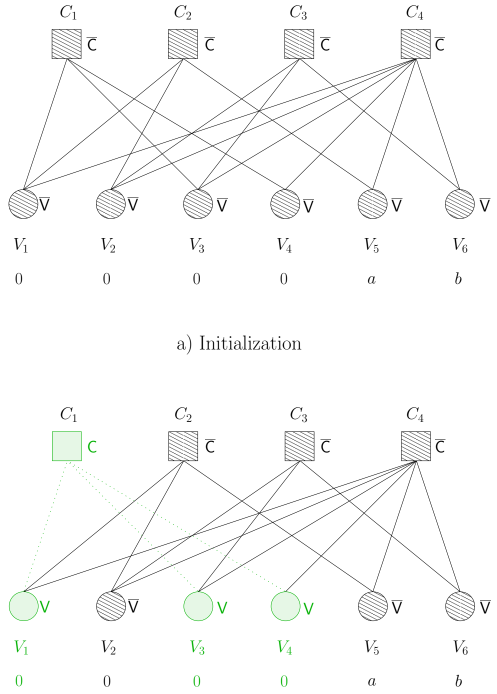
  b. Correction. If all VNs but one connected to a certain CN are verified, the remaining VN is set (according to the parity equation) to the element-wise modulo-2 sum of the verified VNs. Its state is updated to verified (V) and the check node is set to confirmed (C).

Step ii is iterated until no CNs can be confirmed anymore, or when a contradiction due to *false verification* is found. False verifications are induced by two or more simultaneous errors in one equation, yielding anyhow to a valid configuration. We neglect false verification events since they are very unlikely already for moderate packet sizes. For instance, for $l = 30$ bits a CN may erroneously verify a VN with probability 2^{-30}.

VBD may stop prematurely whenever the VNs belonging to a stopping set [18] remain unverified and no CNs connected to the stopping set can be confirmed anymore. Decoding may be resumed by tackling the problem from a different viewpoint. For this purpose, we consider all unverified VNs as erasures that we attempt to recover using MAP erasure decoding. For the sake of clarity the proposed enhanced verification-based decoding (EVBD) algorithm is exemplified hereafter.

Example 3. *Consider the transmission over a q-SC with a $(6, 2)$ code having parity-check matrix*

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$



a) Initialization

a) First iteration

Figure 34: Instance of VBD according to Example 1.

Assume that the all-zero codeword $\mathbf{x} = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$ *is transmitted. The received vector is* $\mathbf{y} = \mathbf{x} + \mathbf{e}$*, where* $\mathbf{e} = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & a & b \end{array}\right]$ *is the error pattern with* $a, b \in \mathbb{F}_q^*$ *and* $a \neq b$*. By* $\mathbb{F}_q^*$ *we refer to the order-*$(q-1)$ *cyclic multiplicative group of* $\mathbb{F}_q$*. The first iteration of Algorithm A is illustrated in Figure 34. Immediately after that, the decoder stops since there is no CN whose parity is satisfied or that is connected to a single unverified VN. Before proceeding with EVBD, let's define by* $\tilde{\mathbf{H}}$ *the matrix* $\mathbf{H}$ *depurated by the rows corresponding to confirmed CN after execution of Algorithm A. We obtain*

$$\tilde{\mathbf{H}} = \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right].$$

In the received vector, we first set the entries associated with verified VNs to the corresponding (possibly corrected) value. We then mark the entries associated with unverified VNs as unknowns which yields $\hat{\mathbf{y}} = \left[\begin{array}{cccccc} 0 & \hat{y}_2 & 0 & 0 & \hat{y}_5 & \hat{y}_6 \end{array}\right]$. *Assuming no false verifications have taken place, MAP erasure decoding is now applied by forcing* $\hat{\mathbf{y}}\tilde{\mathbf{H}}^T = \mathbf{0}$ *or equivalently*

$$\left[\begin{array}{cccccc} 0 & \hat{y}_2 & 0 & 0 & \hat{y}_5 & \hat{y}_6 \end{array}\right] \left[\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{array}\right] = \left[\begin{array}{ccc} 0 & 0 & 0 \end{array}\right].$$

The system can be solved via GE if the rank of the sub-matrix given by the columns of $\tilde{\mathbf{H}}$ *related to the unverified VNs (thus, rows* $2, 5, 6$ *of* $\tilde{\mathbf{H}}^T$*) is equal to the number of unverified VNs. In our case it is easy to check that GE provides the unique solution* $\hat{y}_2 = 0$*,* $\hat{y}_5 = 0$ $\hat{y}_6 = 0$*.*

A simple improvement to Algorithm A, here referred to as Algorithm B, was proposed in [46] and works as follows. Each unconfirmed CN C_i passes to each neighboring unverified VN V_j a suggested value based on the sum (in $\mathbb{F}_q$) of the values associated with all VNs connected to C_i, except of the value of V_j. Whenever an unverified VN V_j receives the same suggested value on two different edges, or any of the suggestions matches the value incoming from the channel, V_j adopts the suggested value and is marked as verified. The probability of false verification remains 2^{-l}, if no length-4 cycles are present in the graph [85]. In fact, length-4 cycles are particularly harmful when Algorithm B is applied, as illustrated by the following example.

Example 4. *Consider again the Tanner graph in Figure 34. Assume* $\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ *and* $\mathbf{e} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. *After running Algorithm A, VNs* V_1, V_4, V_5 *remain unverified and CNs* C_1, C_2, C_4 *unconfirmed. Applying Algorithm B it happens that* V_4 *receives the suggested value* a *both from* C_1 *and* C_4. *This results in a false verification.*

Evolutions of this algorithm toward list-message passing were proposed in [85, 71, 70]. The extended algorithms provide performance improvements at the expense of a increase of the computational complexity and memory requirements (to store the messages), making the decoder impractical already for moderate-depth exploration of a node neighborhood [70] (although in [85] a method to keep the list size manageable was provided, relying on the replacement of unlikely values with erasures). Note that the final erasure recovery stage proposed in this paper can be used in combination with different verification based algorithms. An application to Algorithm B (as well as to Algorithm A) will be presented next.

4.2.4 Efficient EVBD of LDPC Codes and Numerical Results

For LDPC codes, EVBD can be efficiently implemented by exploiting the sparseness of the LDPC code parity-check matrix. In fact, whenever VBD fails, the following MAP erasure decoding stage can be based on efficient decoding algorithms based on a structured GE [65, 13, 60]. The algorithms in [65, 13, 60] take advantage of the matrix sparseness by representing most of the unknowns as a linear combination of a small number of unknowns, also referred to as *reference variables* or *pivots*. Once the pivots are resolved by GE the remaining unknowns can be simply obtained by fast iterative decoding (i.e., back substitution). Effective pivoting algorithms are described in [60], where also a practical software MAP erasure decoder implementation has been demonstrated for a $(2048, 1024)$ LDPC code with a decoder throughput as high as 1.5 Gbps.

We evaluate next the performance under EVBD of a set of regular LDPC codes over a q-SC with $q = 2^{48}$ (i.e., packets of 48 bits). Let's consider first a $(5, 10)$ regular code with $n = 1200$ and $k = 600$. Figure 35 shows the codeword error rate (CER) versus ϵ for Algorithm A and Algorithm B with and without the additional erasure recovery step. For both algorithms, we can observe an improvement due to EVBD, with the best results for Algorithm B under EVBD. In Figure 36 the performance of different regular $(1200, 600)$ LDPC codes is compared using Algorithm B with and without erasure recovery. The $(3, 6)$ regular LDPC code under EVBD outperforms the $(4, 8)$ and the $(5, 10)$ codes by far. It should be noted that all codes have been designed carefully in order to avoid cycles of length 4. Thanks to this and to the sufficiently large alphabet size ($q = 2^{48}$), no false verifications have been encountered throughout the simulations.

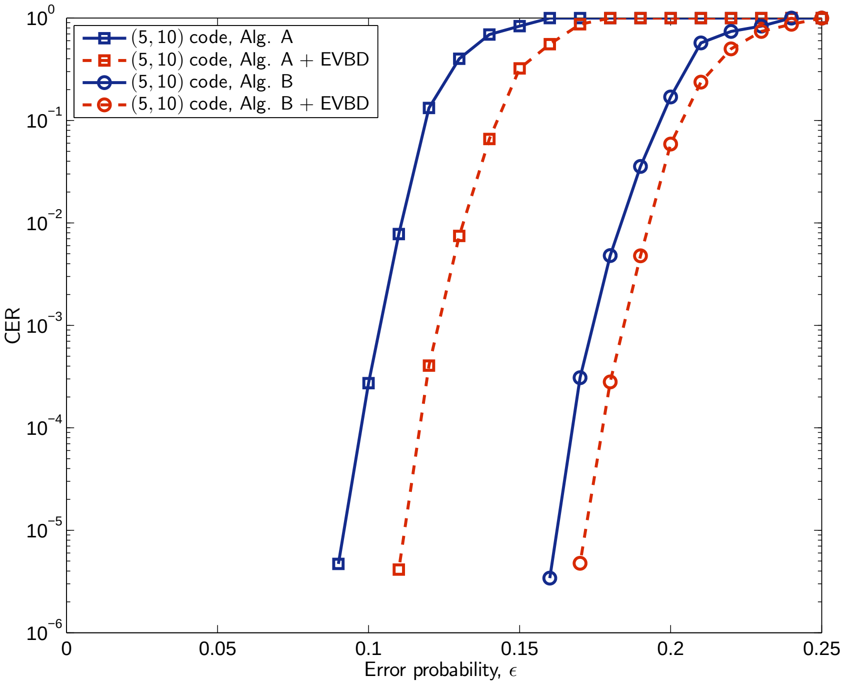


Figure 35: CER vs. error probability ϵ for a $(1200, 600)$ $(5, 10)$ regular LDPC code for Algorithm A and Algorithm B with and w/o erasure recovery step.

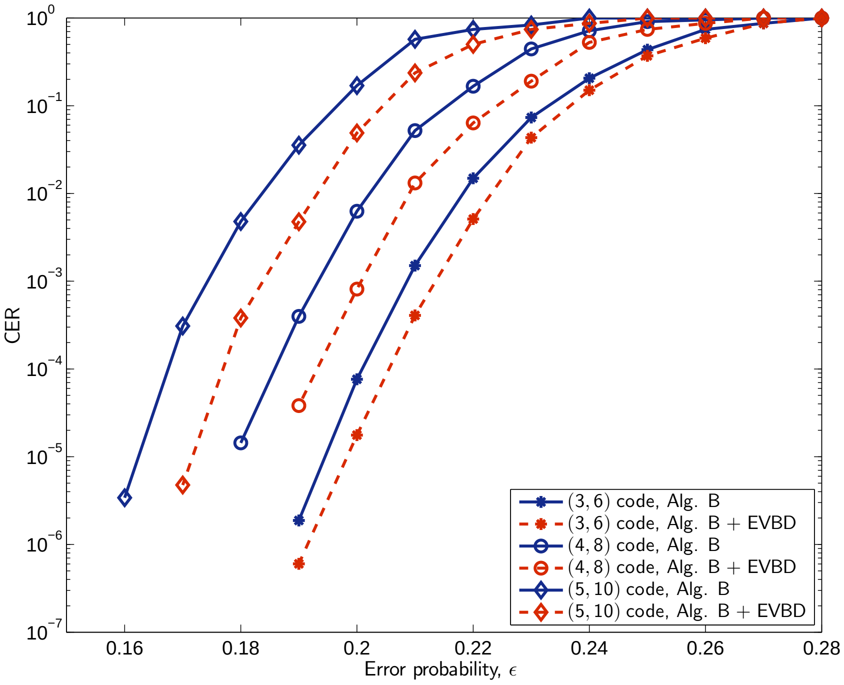


Figure 36: CER vs. error probability ϵ for a $(1200, 600)$ for $(3, 6)$, $(4, 8)$ and $(5, 10)$ regular LDPC codes. Algorithm B with and w/o erasure recovery step.

5 Prioritization for 3D Video over OFDMA

In this section we address 3D video delivery over wireless systems based on OFDMA, by considering a MAC layer scheduling method combined with a prioritized queuing mechanism to prioritize the most important video components/layers with the goal of improving the perceived quality of 3D video at the receiver. We consider colour plus depth 3D video and we exploit its properties in terms of importance of the different components for the perceived quality. The priority values of the SVC encoded 3D video are signalled from the Application Layer to the MAC layer via cross-layer signalling. All the users attached to a Base Station feedback their sub-channel gain to the Base Station periodically via partial Channel State Information (CSI) and this information is used in the sub-channel allocation process at the scheduler. Thereby, the proposed scheduler always guarantees that the most important layers are scheduled over the sub-channels with higher gain at each time slot of an OFDMA frame. Furthermore, we have established a Packet Loss Rate (PLR) threshold which is used by the scheduler to drop video layers which are too much affected by packet losses to save scarce radio resources. Simulation results show that the MAC layer PLR is improved in the prioritized colour/depth layers at the cost of slight increase in PLR in the low prioritized layers. This results in a global quality improvement for the prioritized case. Future extensions of this work include the application and tailoring of the prioritization approach for 3D medical video, also focusing on the exploitation of context information.

5.1 Introduction

3D video delivery over wireless networks is a challenging task due to the inherent characteristics of wireless channels, like limited air resources and time-varying nature, and to the large bandwidth required by 3D video. OFDMA has become the prominent candidate for multimedia delivery in recent years due to its ability to support high data rates over frequency selective fading channels. OFDMA has achieved all these improvements by exploiting multiuser diversity in the MAC layer scheduling and flexible resource allocation (time, frequency and power) at the Physical layer. For this reasons it has been adopted in a number of recent wireless standards, such as IEEE 802.16 (WIMAX), 3GPP LTE, etc.

Both channel and content aware scheduling methods have been investigated for 2D video transmission over an OFDMA based wireless network using cross-layer design [1]-[3]. We extend here the study for 3D video prioritization and scheduling over OFDMA based radio networks; our proposed methodology is applicable for any multi-layer coded video transmission over OFDMA radio networks with pre-assigned priorities.

There are few studies carried out in the literature to prioritize single user data over multiple queues by considering some characteristics of the source content. However, most of them have focused on improving network Quality of Service (QoS) factors like throughput, delay, etc. rather than the QoE of the end user. In particular, the study in [87] addressed batch based scheduling by considering the delay satisfaction and evaluated the performance in terms of system bandwidth efficiency, best-effort (BE) traffic throughput and QoS traffic delay. Furthermore, existing studies are not particularly focused on 3D video delivery.

How scalability information of an SVC bit stream can be used in scheduling and resource allocation for OFDM downlink systems has been investigated in several studies including [33]. To the best of our knowledge none of them have considered 3D video representations, and also most of them have considered only a single queue per user. Here we consider a colour plus depth represented 3D video stream encoded using parallel SVC encoders and prioritized at the MAC layer based on the Dependency/Temporal/Quality (DTQ) ID's of the SVC bit stream.

Depth Image-Based-Rendering (DIBR) is a commonly used 3D video rendering method in research and standardization activities due to its simplicity and backward compatibility [30]. In DIBR, the colour video sequence has a higher importance than the depth map sequence towards the perceived 3D video quality after rendering. The importance of prioritizing colour video data over depth map information is highlighted in [31]. By considering these facts, we have encoded the colour plus depth video components in their rendering order and these layer arrangements are signalled to the MAC layer using cross-layer messages and then utilized in the scheduling decisions.

The remainder of this Section is organized as follows. Subsection 5.2 presents the proposed prioritization and scheduling approach; Subsection 5.3 presents the simulation set-up considered in the experiments, whereas numerical results are presented in Subsection 5.4 and conclusions are drawn in Subsection 5.5.

5.2 Proposed content and channel-aware prioritization and scheduling strategy

We assume a video transmission scenario where K users are receiving from the access point/base station/eNodeB K stereoscopic video streams encoded through a scalable encoding scheme in N layers. The wireless system considered is a generic system based on OFDMA, such as mobile WiMAX and LTE/LTE-A systems.

The structure of the OFDMA transmission frame is depicted in Fig. 37. It is composed of M_s OFDM symbols and M_{sc} sub carriers. The minimum amount of radio resources that can be allocated to a user at time slot i and sub channel j is the resource block $RB(i,j)$, where $i = \{1, 2, ..., N_{ts}\}$ and $j = \{1, 2, ..., N_{sch}\}$. N_{ts} and N_{sch} are the no. of time slots and the no. of frequency sub-channels of the OFDMA frame respectively.

Each RB consists of RB_{sc}=M_{sc}/N_{sch} sub carriers and RB_s=M_s/N_{ts} OFDM symbols.

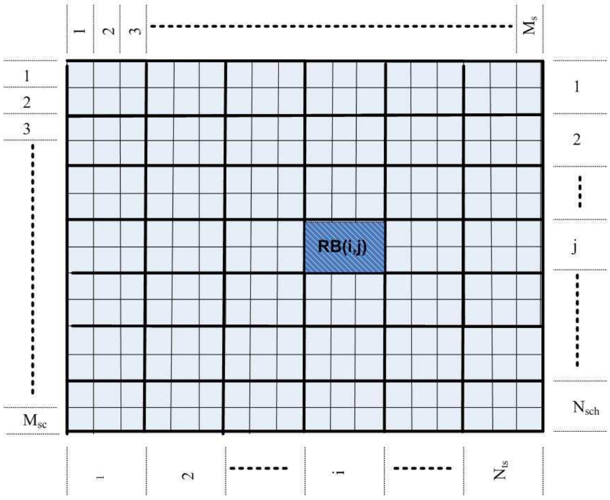


Figure 37: Structure of OFDMA transmission frame

General MAC implementations have only a single queue for each user/traffic flow and packets coming from the upper layers are scheduled on a first come first served basis. In order to add a higher flexibility in the scheduling process, with the knowledge of the importance of a particular video packet towards the perceived 3D video quality at the receiver, we introduce here a prioritized queue system for each user. We refer to these two queuing methods as Legacy and Prioritized MAC Queuing in the rest of the paper and their operation is depicted in Fig. 38.

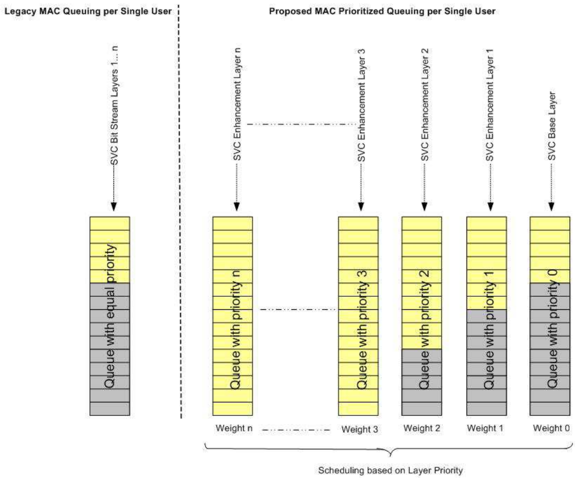


Figure 38: Legacy vs. prioritized queuing

5.2.1 Weight Adaptation

A weight adaptation algorithm to schedule heterogeneous traffic while maximizing the Weighted Sum Rate (WSR) by scheduling transmission and allocating Resource Blocks (RB) accordingly was proposed in [32].

We adopt here a modified version of this algorithm for Multi-Layer Coded (MLC) video users: here each user has several prioritized queues in the proposed scheduling scenario. We propose to add content awareness to the weight adaptation algorithm by considering the fractional rate of the layer and time-to-deadline of the Head-Of-Line (HOL) packet, which is the first packet waiting for transmission in each queue, to account for the peaks in source rate. In addition to that, we propose to control the layer priority explicitly by queuing the layers based on their priority, signaled through cross-layer signaling, by making sure that the highly prioritized queue is scheduled first on the sub-channel with higher gain at each scheduling interval. The weight for each user k is selected based on the proposed adaptation algorithm given in (1):

$$W_k = \begin{cases} A_k e^{dT_k} & \text{if } k \text{ is RT} \\ A_k e^{dT_k^{max}} (1 + a(1 - B_{k,i}) e^{-bTD_{k,i}/J_{k,i}}) & \text{if } k \text{ is MLC RT} \\ A_k P_k & \text{if } k \text{ is NRT} \end{cases} \tag{18}$$

where d is a proper constant, whose value can be adjusted to tune the average latency experienced by the RT users, T_k is the time spent in queue by the HOL packet for user k, T_k^{max} is the time spent in the queue by the "eldest" HOL packet among the queues for user k, a and b are used to suitably shape this function of the time-to-deadline $TD_{k,i}$ (of the HOL packet), $J_{k,i}$ is the maximum tolerable delay of the application based on the payload time (frame rate), P_k is another constant representing the initial priority value for the NRT user k and, finally, A_k is a dynamically updated parameter controlling the throughput fairness among the users.

To guarantee fairness of throughput among the users, the authors in [32] proposed a low-complexity algorithm which iteratively updates the A_k's based on the achieved throughput and the target throughput of each user. $B_{k,i}$ is the fractional rate, defined as follows:

$$B_{k,i} = B_i / \sum_{i=1}^{N} B_i$$

with

B_i - Video layer bitrate
N - Total numbers of video layers

The difference between the RT and the NRT weights is clear: in order to guarantee a limited transmission latency, the RT priority is exponentially increasing with T_k. Hence the resource allocator will tend to favour the RT users with packets waiting in queue for a longer time. For Multi-Layer Coded video RT users, the longest waiting HOL packet is considered among several prioritized queues and the weight is shared proportionally to the fractional rate of the layer queued on that particular queue. Apart fror that, the weight W_k increases exponentially as the TD_k decreases.

The parameter settings obtained via numerical analysis and used in this study are d=1, a=4, b=2, $J_{k,i}$=1/25s (1/12.5s) as sequences at frame rate of 25fps (12.5fps) are used ,and P_k=0.1.

Due to the proposed prioritized queuing, packets may arrive out of order at the receiving node, hence packets are re-ordered using the *Sequence Number* header field in Frame Control of MAC header before forwarding to the upper layers.

5.2.2 Allocation Strategies

The three subsections below describe how subcarriers and power are allocated among the different priority queues and how many bits are loaded on each RB at each time slot. A number of works have considered the maximization of WSR on multiuser OFDM systems through optimal sub-carrier and power allocation. In particular, [10] proved that exclusive subcarrier assignment (i.e., each sub-carrier is allocated to only one user), along with an appropriate power allocation, maximises the WSR in the case of user wise target Bit Error Rate (BER)/Frame Error Rate (FER).

5.2.3 Subcarrier Assignment

Sub-carriers are assigned to the prioritized queues in order of their channel gain obtained via the feedback CSI, i.e., the queue with the highest priority is scheduled on the sub-channels with the highest gain and then the others are assigned in order of their priority [32]-[25]. How multimedia transmission over wireless medium can be improved with partial CSI is described in [25].

5.2.4 Bit Loading

Bit loading for the scheduled layer is performed by considering the channel gain feedback sent by a particular user via CSI as described in [32] [25]. The no. of bits allocated per Resource Block $RB(i,j)$ for the $k-th$ Priority Queue is:

$$C_k(n, P_k(i,j)) = RB_{sc}RB_s \log_2 (1 + G_n P_k(i,j)) \qquad [bits/RB] \tag{19}$$

with

$P_k(i,j)$ - Power allocated to each subcarrier of $RB(i,j)$

G_n - Normalized SNR for subcarriers belongs to subchannel n of Priority Queue k

and RB_{sc} and RB_s defined above.

5.2.5 Power Allocation

In the simulations, the power was allocated using the efficient multi-user water filling algorithm proposed in [57] [58]. The transmission power for the $k-th$ user on the $j-th$ subcarrier is given as in (3).

$$P_{k,j} = RB_{sc}RB_s(2^c - 1)F_k/SNR_j \tag{20}$$

SNR gap,

$$F_k = \left[Q^{-1}(P_e/4)\right]^2 \tag{21}$$

with

SNR_j - Signal to Noise Ratio (SNR) of $j-th$ subcarrier

c - coded/uncoded bits to be transmitted

Q^{-1} - Inverse complementary error function

P_e - Bit Error Probability

The SNR gap in the above equation can be interpreted as a link between the information theoretic channel capacity and the bitrate which is achievable with the chosen modulation scheme.

5.3 Simulation set-up

This section summarizes the simulation setup used in our study. We consider the *Orbi* test sequence in colour plus depth format. A three seconds long sequence, (*i.e.*, 75 frames from the *Orbi* sequence at 25 fps) is encoded with *Intra Period* of 25 frames. 33 macroblocks (*MB*) per slice are also introduced in order to make the decoding process more robust to errors. The colour and depth components are encoded in two different spatial resolutions (176x144, 352x288) with different compression ratios (QP 36, 30) using the JSVM 9.19.8 reference software encoder implementation for comparative performance analysis.

We used symmetrical coding, i.e.,both the colour sequence and the associated depth map are encoded with the same level of compression (QP). Moreover, we organized the layers in the order of DIBR rendering, with the colour sequence prioritized with respect to the associated depth map, by considering their importance towards the rendered video quality [31]. The SVC layer arrangement is shown in Table I, where DTQ are Dependency, Temporal, and Quality ID's of the SVC bit stream respectively.

Table 8: SVC LAYER ARRANGEMENT

Layer	Resolution	Frame rate	Bit rate	DTQ	Description
0	176x144	12.5	36.5	0,0,0	Base Layer (Color, 176x144, QP=36)
1	176x144	25	49.3	0,1,0	Enhancement Layer1 (Color, 176x144, QP=36)
2	176x144	12.5	96.0	0,0,1	Enhancement Layer2 (Depth, 176x144, QP=36)
3	176x144	25	122.5	0,1,1	Enhancement Layer3 (Depth, 176x144, QP=36)
4	352x288	12.5	337.9	1,0,0	Enhancement Layer4 (Color, 352x288, QP=30)
5	352x288	25	589.1	1,1,0	Enhancement Layer5 (Color, 352x288, QP=30)
6	352x288	12.5	851.8	1,0,1	Enhancement Layer6 (Depth, 352x288, QP=30)
7	352x288	25	1188.5	1,1,1	Enhancement Layer7 (Depth, 352x288, QP=30)

OFDMA parameter settings used in the simulations are reported in the Table II.

User Datagram Protocol (UDP) was used as the transport protocol and Automatic Repeat reQuest (ARQ) is disabled at the MAC layer. Hence, no packets with bit errors are received by the application layer at the receiver. Slice Interpolation was used as the error concealment technique at the decoder, which is a slightly modified version of the JSVM decoder defined in [12].

We analyzed the performance of our proposed methodology with respect to the MAC scheduling scheme in [32] ("Legacy"). In the Legacy MAC scheduling scheme, there is only a single queue for a particular user while packets from different layers are scheduled on a first come first served basis, without considering any importance of the packet towards the perceived video quality.

We evaluated the improvement in performance with the ratio of MAC data packets that are discarded in each layer due to CRC failures when both Prioritized and Legacy MAC is in use. The layer information is signalled to the receiving end via the MAC layer QoS Markings and Class Of Service (COS) to support the calculation of Frame Losses in each layers at the receiver side.

We are also evaluating the quality improvement in terms of Peak Signal to Noise Ratio (PSNR) of individual

Table 9: OFDM PARAMETERS

No. of Tx Antennas	1
No. of Rx Antennas	1
No. of Carriers (M_{sc})	48
No. of Pilot Carriers	4
No. of Total Carriers	64
No. of Sub Channels (N_{sch})	12
No. of Symbols (M_s)	96
No. of Time Slots (N_{ts})	8
Total Bandwidth	20 Mbps
Coherence Bandwidth	1.25 Mbps
Coherence Time	0.1 s
Shadowing Time	3 s
Standard Deviation	4 dB

SVC scalability layers as well as DIBR rendered left and right video sequences.

5.4 Simulation results and discussion

Fig.39 illustrates the average video quality variation of individual layers when prioritized MAC scheduling (PMAC) is in use compared to legacy MAC (LMAC) scheduling.

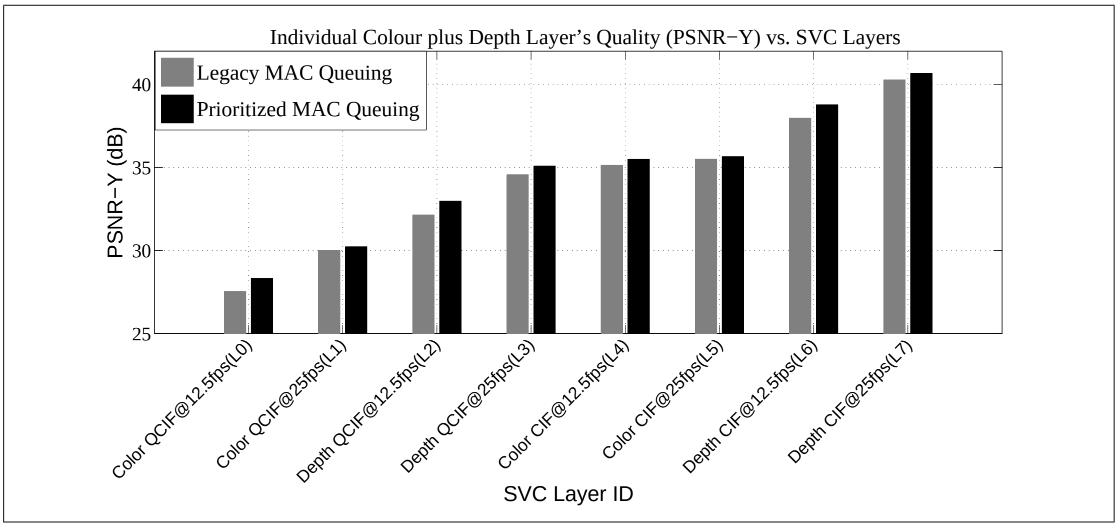


Figure 39: Video quality (PSNR) vs. SVC layer ID's

We also evaluated the video quality of the stereoscopic views rendered for 3D video displays using the Depth Image Based Rendering (DIBR) method proposed in [73]. The relevant results are reported in Fig. 40.

As it can be observed from Fig.39, improvement in the video quality of almost all the colour plus depth prioritized layers can be achieved in the proposed prioritized queuing compared to the reference legacy MAC queuing.

Table 10 reports the MAC data packets drop ratios of individual layers when prioritized MAC scheduling (PMAC) is in use compared to legacy MAC (LMAC) scheduling. According to Table 10, MAC packet drop ratios are almost equal for all the layers in legacy MAC queuing for a given E_S/N_0. For the prioritized queuing, MAC packet drop ratios are improved in the prioritized layers while layers with low priority are having higher packet drop ratios compared to the legacy queuing. The reason for this behaviour is that the layers with higher priority are less prone to bit errors introduced at the channel, because they are always scheduled on the channels with higher gain and vice versa.

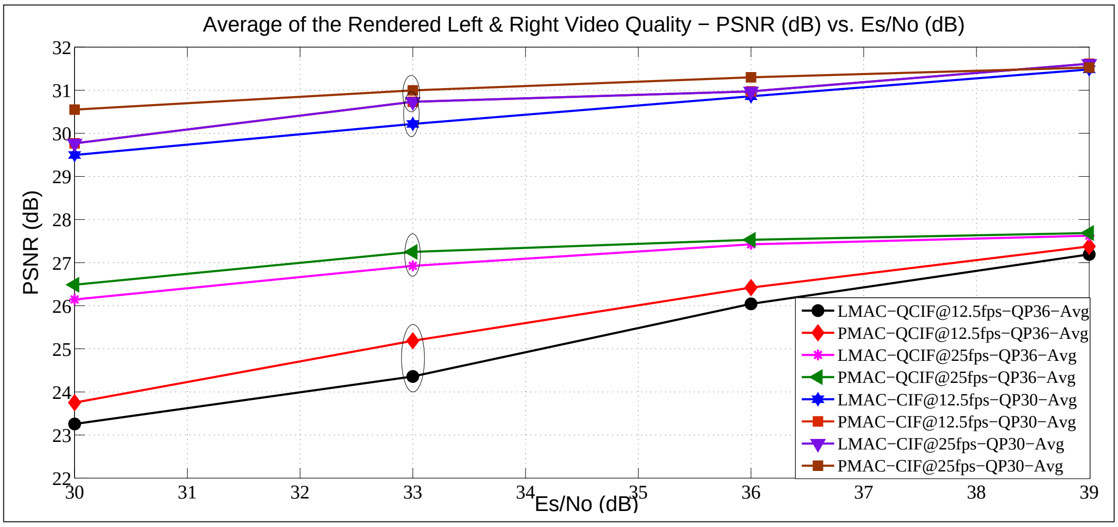


Figure 40: Averaged left and right video quality - PSNR vs. Es/No

Table 10: MAC DATA PACKETS DROP RATIOS

E_S/N_0 (dB)	L/P MAC	MAC Packets Dropped Ratio (%)							
		L0	L1	L2	L3	L4	L5	L6	L7
39	LMAC	1.32	1.35	1.75	2.70	1.54	1.58	1.54	1.58
	PMAC	0.44	0.45	0.88	1.80	1.32	1.58	1.75	1.58
36	LMAC	3.07	2.70	3.51	3.60	3.51	3.60	3.95	4.05
	PMAC	2.19	1.80	2.63	2.70	3.29	3.38	4.17	4.28
33	LMAC	5.70	4.95	5.26	5.41	5.26	5.86	5.92	5.86
	PMAC	3.95	4.05	4.39	4.50	5.04	5.63	5.70	6.08
30	LMAC	8.33	8.11	8.77	8.11	8.55	8.33	7.89	8.56
	PMAC	6.14	6.76	7.02	7.21	7.89	7.88	7.68	8.33
Average	LMAC	4.61	4.28	4.82	4.95	4.71	4.84	4.82	5.01
	PMAC	3.18	3.27	3.73	4.05	4.39	4.62	4.82	5.07

Fig.40 shows that the rendered left and right views always achieves better video quality when the proposed prioritized queuing is used compared to the legacy queuing. As it can be seen from Table 10 and Fig. 39, video quality has been improved in the less prioritized layers (L6, L7) though they have undergone high packet drop ratio when using prioritized MAC. This is due to the fact that any quality improvement in a lower layer of the prediction hierarchy of SVC can improve the quality of higher layers, which are inter-predicted from that particular layer.

We analysed the video quality variation of the Rendered Left and Right Video Quality for different PLR and results are reported in Fig. 41. According to Fig. 41, we can observe a drastic drop in the video quality when PLR exceeds 2% in all the layers.

Based on these observations we can establish a PLR threshold, such that when the packet loss ratio is above a threshold value, it is more convenient to fully drop the layer than to consider the one affected by losses and to perform error concealment. Such a threshold value depends on the concealment strategy adopted.

5.5 Conclusion

Considering the above observations, we can conclude that the proposed prioritized scheduler always guarantees an improvement in the packet drop ratio of high priority layers by allocating them to the sub-channels with higher gain. More importantly this improvement is achieved with a slight increase in packet drop ratio in the less perceptually important video layers. The proposed scheduler also improves the utilization of the scarce radio resources and reduces the burden of error concealment due to lost video packets, by discarding low prioritized video layers, if they exceed the established PLR threshold at the base station MAC.

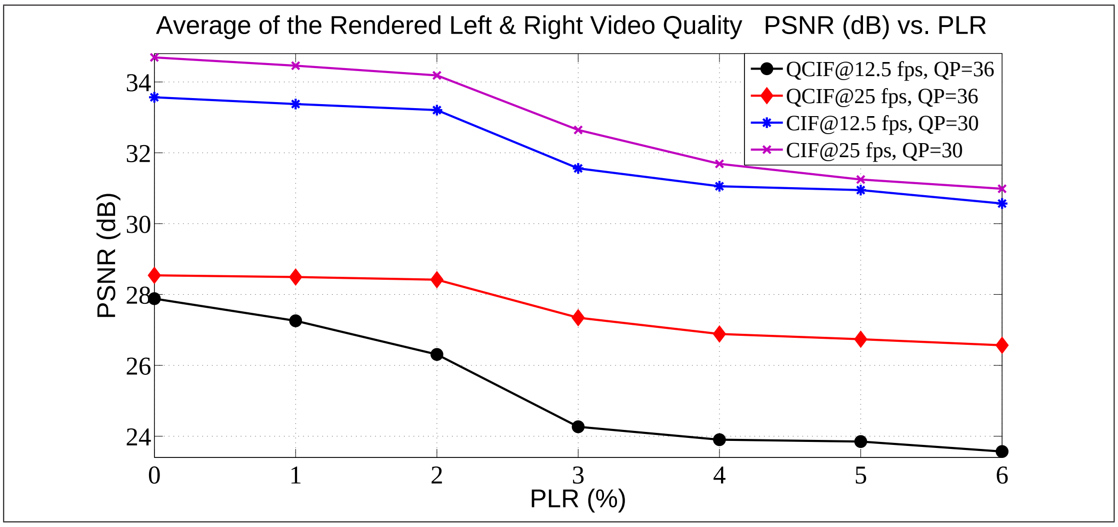


Figure 41: PSNR vs. Packet Loss Ratio

6 Conclusion

In this deliverable several context-awareness techniques have been described and evaluated, through the CONCERTO content transmission architecture. This work on context-awareness techniques will be the starting point for the work that will continue throughout the WP4 lifetime and that will be presented in two more subsequent deliverables (D4.2 and D4.3). In this document several descriptions and evaluations of context-awareness techniques have been proposed, capable to cope with different healthcare and safety use case scenarios and some preliminary scientific results have been provided. These results will be demonstrated in WP6 together with techniques developed in other work packages in the CONCERTO simulation chain based on OMNeT++ and could be tested in the CONCERTO demonstrator.

7 Acronyms

ARQ	automatic repeat request
BEC	binary erasure channel
BLEC	block erasure channel
BP	belief propagation
CER	codeword error rate
CN	check node
CRC	cyclic redundancy check
EVBD	enhanced verification-based decoding
FoV	field of view
GE	Gaussian elimination
LDPC	low-density parity-check
LT	Luby transform
MAP	maximum a-posteriori
MDS	maximum distance separable
ML	maximum likelihood
MSHTM	multi-source and holographic tele-medicine
PPM	pulse position modulation
QoE	quality of experience
QoS	quality of service
RTP	real-time transport protocol
VBD	verification-based decoding
VN	variable node
VSN	visual sensor network
SVC	Scalable Video Coding
RTP	real time transport protocol
LDPC	low density parity check
AWGN	additive white Gaussian noise
QoS	Quality of Service
PSNR	Peak Signal-to-Noise Ratio
CSI	Channel State Information
FEC	forward error correction
OFDMA	Orthogonal Frequency Division Multiple Access
PLR	packet loss rate
RTT	round trip time . . .
UDP	User Datagram Protocol
ARQ	automatic retransmission request
PLR	Packet Loss Rate
RTT	round trip time
BER	Bit Error Rate
MAC	Medium Access Control
UDP	User Datagram Protocol
HARQ	Hybrid Automatic Retransmission reQuest
FER	Frame Error Rate
CRC	Cyclic Redundancy Check
QoE	Quality of Experience
WSR	Weighted Sum Rate
DASH	Dynamic Adaptive Streaming over HTTP
CDN	Content Delivery Network
XML	Extensible Markup Language
HLS	HTTP Live Streaming
IIS	Microsoft Internet Information Services
MPD	Media Presentation Description

MVC	Multiview Video Coding
SVC	Scalable Video Coding
URL	Uniform Resource Locator
HDS	HTTP Dynamic Streaming
QoE	Quality-of-Experience
RTT	Round-Trip Time

References

[1] 3gp-dash: http://www.3gpp.org/ftp/specs/html-info/26247.htm, Access date: June 2012.

[2] Future of html5 and http streaming: http://www.streamingmediaglobal.com/articles/editorial/featured-articles/google-outlines-crucial-html5-video-development-during-keynote-82632.aspx, Access date: June 2012.

[3] Http dynamic streaming: http://www.adobe.com/products/hds-dynamic-streaming/features.html, Access date: June 2012.

[4] Http live streaming: https://developer.apple.com/resources/http-streaming/, Access date: June 2012.

[5] Mpeg-dash: http://dashpg.com/, Access date: June 2012.

[6] Smooth streaming: http://www.iis.net/download/smoothstreaming, Access date: June 2012.

[7] T. Banka A. Bare A.Jayasumana, N. Piratla and R. Whitner. *RFC 5236 Improved Packet Reordering Metrics*. 2008.

[8] S. Akhshabi, L. Anantakrishan, C. Dovrolis, and A.C. Begen. What happens when http adaptive streaming players compete for bandwidth. In *Proceedings of NOSSDAV12*, June 2012.

[9] S. Akhshabi, A.C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In *Proceedings of MMSys11*, February 2011.

[10] A. Alsawah and I. Fajalkow. Weigthed sum-rate maximization in multiuser-OFDM systems under differentiated quality-of-service constraints. In *IEEE Workshop on Signal Processing Advances in Wireless Communications*, Helsinki, Jun. 2007.

[11] E.R. Berlekamp. The technology of error-correcting codes. *Proc. IEEE*, 68(5):564–593, May 1980.

[12] M. B. Brandas, M. Uitto, M. G. Martini, and J. Vehkapera. Quality assessment and error concealment for SVC transmission over unreliable channels. In *IEEE International Conference on Multimedia and Expo*, Barcelona, Jul. 2011.

[13] D. Burshtein and G. Miller. An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel. *IEEE Trans. Inf. Theory*, 50(11):2837–2844, November 2004.

[14] J.W. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to reliable distribution of bulk data. *IEEE J. Sel. Areas Commun.*, 20(8):1528–1540, October 2002.

[15] M. Chiani, M. Mazzotti, and S. Moretti. Camera ranking selection algorithms for large area monitoring. *CNIT, University of Bologna*, 2012. to be submitted.

[16] L. De Cicco, S. Mascolo, and V. Palmisano. Feedback control for adaptive live video streaming. In *Proceedings of MMSys11*, February 2011.

[17] M. Cunche, V. Savin, and V. Roca. Analysis of quasi-cyclic LDPC codes under ML decoding over the erasure channel. In *Proc. of Int. Symp. Inf. Theory and its Applications*, pages 861–866, Taichung, Taiwan, October 2010.

[18] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke. Finite length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inf. Theory*, 48:1570–1579, June 2002.

[19] D. Divsalar, S. Dolinar, and C. Jones. Protograph LDPC codes over burst erasure channels. In *Proc. of 2006 IEEE Military Commun. Conf.*, pages 1–7, Washington, DC, USA, October 2006.

[20] P. Elias. Coding for two noisy channels. In *In Proc Information Theory: Third London Symposium*, pages 61–74. London: Butterworth Scientific, Ed. C. Cherry, 1955.

[21] A. Ercan, D. Yang, A. El Gamal, and L. Guibas. Optimal placement and selection of camera network nodes for target localization. In *Distributed Computing in Sensor Systems*, Lecture Notes in Computer Science, pages 389–404 vol. 4026. Springer Berlin / Heidelberg, 2006.

[22] K. Evensen et al. Using bandwidth aggregation to improve the performance of quality-adaptive streaming. *Signal Processing: Image Communication*, pages 312–328, 2012.

[23] Shervan Fashandi, Shahab Oveis Gharan, and Amir K. Khandani. Coding over an erasure channel with a large alphabet size. In *Proc. 2008 IEEE Int. Symp. Inf. Theory*, pages 1053–1057, Toronto, Canada, July 2008.

[24] A. Fecheyr-Lippens. A review of http live streaming: http://files.andrewsblog.org/http_live_streaming.pdf, Access date: July 2010.

[25] R. Fracchia, C. Lamy-Bergot, G. Panza, J. Vehkapera, E. Piri, T. Sutinen, M. Mazzotti, M. Chiani, S. Moretti, G. Jeney, L. Bokor, Z. Kanizsai, and M. G. Martini. System architecture for multimedia streaming optimisation. In *Future Network and Mobile Summit 2010*, Florence, Italy, Jun. 2010.

[26] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, MA, 1963.

[27] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[28] A. Guillen i Fabregas. Coding in the Block-Erasure Channel. *IEEE Trans. Inf. Theory*, 52(11):5116 –5121, November 2006.

[29] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan. Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *Proc. of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pages 360–369, Cambridge,Massachusetts, USA, 2007. ACM Press.

[30] C. Hewage, S. Nasir, S. Worrall, and M. G. Martini. Prioritized 3D video distribution over IEEE 802.11e. In *Future Network and Mobile Summit*, Florence, Italy, Jun. 2010.

[31] C. T. E. R. Hewage, S. Worrall, S. Dogan, H. K. Arachchi, and A. M. Kondoz. Stereoscopic TV over IP. In *Proc of the 4th IET European Conference on Visual Media Production (CVMP'2007)*, London, UK, Nov. 2007.

[32] D2.3c ICT FP7 OPTIMIX project. *JSCC Controller: Final Design and Algorithmic Optimization*. 2010.

[33] X. Ji, J. Huang, M. Chiang, G. Lafruit, and F. Catthoor. Scheduling and resource allocation for SVC streaming overOFDM downlink systems. *IEEE Transactions on Circuits and Systems for Video Technology 19 (10)*, 19(10):1549–1555, Oct. 2009.

[34] T. H. Ko and N. M. Berry. On scaling distributed low-power wireless image sensors. In *Proc. of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, pages 235.3– vol.9, Washington, DC, USA, 2006.

[35] C. Koller, M. Haenggi, J. Kliewer, and D.J. Costello, Jr. On the optimal block length for joint channel and network coding. In *Proc. of 2011 IEEE Inf. Theory Workshop*, pages 528–532, Paraty, Brazil, October 2011.

[36] Georg Lansberg. Uber eine anzahlbestimmung und eine damit zusammenhangende reihe. *Journal fur die reine und angewandte Mathematic*, 3:87–88, 1893.

[37] M. Lentmaier, A. Sridharan, D.J. Costello, Jr., and K.Sh. Zigangirov. Iterative decoding threshold analysis for LDPC convolutional codes. *IEEE Trans. Inf. Theory*, 56(10):5274–5289, October 2010.

[38] R. Lidl and H. Niederreiter. *Finite fields*. Cambridge Univ. Press, 1997.

[39] C. Liu, I. Bouazizi, and M. Gabbouj. Rate adaptation for adaptive http streaming. In *Proceedings of MMSys11*, February 2011.

[40] G. Liva, E. Paolini, and M. Chiani. Performance versus overhead for fountain codes over $\mathbb{F}_q$. *IEEE Commun. Lett.*, 14(2):178–180, February 2010.

[41] G. Liva, E. Paolini, B. Matuz, and M. Chiani. A decoding algorithm for LDPC codes over erasure channels with sporadic errors. In *Proc. 48th Allerton Conf. Commun., Control, and Computing*, pages 458–465, Monticello, IL, USA, September 2010.

[42] T. Lohmar, T. Einarsson, P. FrÃ¶jdh, F. Gabin, and M. Kampmann. Dynamic adaptive http streaming of live content. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM) IEEE International Symbosium*, June 2011.

[43] M. Luby. LT codes. In *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 271–282, Vancouver, Canada, November 2002.

[44] M. Luby, W. Law, T. Fautier, M. Watson, D. Price, and I. Sodagar. Mpeg-dash: Driving the growth of streaming using the new http standard. In *Streaming Media West*, Nov. 2011.

[45] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proc. 29th Symp. Theory Computing*, pages 150–159, 1997.

[46] M.G. Luby and M. Mitzenmacher. Verification-based decoding for packet-based low-density parity-check codes. *IEEE Trans. Inf. Theory*, 51(1):120–127, January 2005.

[47] Michael Luby, Michael Mitzenmacher, Amin Shokrollahi, and Daniel A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Trans. Inf. Theory*, 47(2):585–598, February 2001.

[48] D.E. Lucani, M. Médard, and M. Stojanovic. Random linear network coding for time-division duplexing: Field size considerations. In *Proc. of 2009 IEEE Global Telecommun. Conf.*, pages 1–6, Honolulu, HI, USA, November/December 2009.

[49] F.J. Mac Williams and N.J.A. Sloane. *The theory of error-correcting codes*, volume 16. North Holland Mathematical Libray, 1977.

[50] S.J. MacMullan and O.M.Collins. A comparison of known codes, random codes, and the best codes. *IEEE Trans. Inf. Theory*, 44, November 1998.

[51] J. Massey. Capacity, cutoff rate, and coding for a direct-detection optical channel. *IEEE Trans. Commun.*, 29(11):1615–1621, November 1981.

[52] R. McEliece. Practical codes for photon communication. *IEEE Trans. Inf. Theory*, 27(4):393–398, July 1981.

[53] J. Metzner. Majority-logic-like decoding of vector symbols. *IEEE Trans. Commun.*, 44(10):1227–1230, October 1996.

[54] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz. Adaptation algorithm for adaptive streaming over http. In *Proceedings of 2012 IEEE 19th International Packet Video Workshop*, May 2012.

[55] R.K.P. Mok, X. Luo, E.W.W. Chan, and R.K.C. Chang. Qdash: A qoe-aware dash system. In *Proceedings of MMSys12*, February 2012.

[56] K. Morioka and H. Hashimoto. Appearance based object identification for distributed vision sensors in intelligent space. In *Proc. of Intelligent Robots and Systems (IROS '04). IEEE/RSJ International Conference on*, pages 199–204 vol.1, Sendai, Japan, 2004.

[57] C. Muller, A. Klein, F. Wegner, M. Kuipers, and B. Raaf. Dynamic subcarrier, bit and power allocation in OFDMA-based relay networks. In *Proc of 12th International OFDM-Workshop*, Hamburg, Germany, Aug. 2007.

[58] G. Munz, S. Pfletschinger, and J. Speidel. An efficient water-filling algorithm for multiple access OFDM. In *IEEE Global Telecommunications Conference*, pages 681–685, Taipei, Nov. 2002.

[59] K. Obraczka, R. Manduchi, and J.J. Garcia-Luna-Aveces. Managing the information flow in visual sensor networks. In *Proc. of the 5th Wireless Personal Multimedia Communications (WPMC '02), International Symposium on*, pages 1177–1181 vol.3, Sheraton Waikiki, Honolulu, Hawaii, 2002.

[60] E. Paolini, G. Liva, B. Matuz, and M. Chiani. Maximum likelihood erasure decoding of LDPC codes: Pivoting algorithms and code design. *IEEE Trans. Commun.* to appear.

[61] Enrico Paolini, Gianluigi Liva, Balazs Matuz, and Marco Chiani. Generalized IRA erasure correcting codes for hybrid Iterative/Maximum Likelihood decoding. *IEEE Commun. Lett.*, 12(6):450–452, June 2008.

[62] J. Park, P. Bhat, and A. Kak. A look-up table based approach for solving the camera selection problem in large camera networks. In *Proc. of the International Workshop on Distributed Smart Cameras (DCS '06)*, 2006.

[63] H. D. Pfister, I. Sason, and R. Urbanke. Capacity-achieving ensembles for the binary erasure channel with bounded complexity. *IEEE Trans. Inf. Theory*, 51(7):2352–2379, July 2005.

[64] N. M. Piratla and A. P. Jayasumana. Metrics for packet reordering , a comparative analysis. *International Journal of Communications Systems*, 21(1):99–113, Jan 2007.

[65] H. Pishro-Nik and F. Fekri. On decoding of low-density parity-check codes over the binary erasure channel. *IEEE Trans. Inf. Theory*, 50(3):439–454, March 2004.

[66] V. Roca, C. Neumann, and D. Furodet. Low density parity check (LDPC) staircase and triangle forward error correction (FEC) schemes. Request for comment 5170 (“Standards Track/Proposed Standard”), IETF RMT Working Group, June 2008.

[67] B. Schotsch, R. Lupoaie, and P. Vary. The performance of low-density random linear fountain codes over higher order Galois fields under maximum likelihood decoding. In *Proc. of the 49th Allerton Conf. Communication, Control, and Computing*, pages 1004–1011, Monticello, IL, USA, September 2011.

[68] B. Schotsch, H. Schepker, and P. Vary. The performance of short random linear fountain codes under maximum likelihood decoding. In *Proc. of 2011 IEEE Int. Conf. Commun.*, pages 1–5, Kyoto, Japan, June 2011.

[69] A. Shokrollahi. Capacity-approaching codes on the q-ary symmetric channel for large q. In *Proc. 2004 IEEE Inf. Theory Workshop*, pages 204–208, San Antonio, TX, USA, October 2004.

[70] A. Shokrollahi and W. Wang. Capacity approaching low-density parity-check codes on the the q-ary symmetric channel for large q. unpublished, 2004.

[71] A. Shokrollahi and W. Wang. Low-density parity-check codes with rates very close to the capacity of the q-ary symmetric channel for large q. In *Proc. 2004 IEEE Int. Symp. Inf. Theory*, page 275, Chicago, IL, USA, June/July 2004.

[72] M.A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551–2567, June 2006.

[73] D. De Silva, W. Fernando, and H. Kodikaraarachchi. A new mode selection technique for coding depth maps of 3D video. In *IEEE International Conference on Acoustics*, pages 686–689, Dallas, Mar. 2010.

[74] Iraj Sodagar. Industry and standards: The mpeg-dash standard for multimedia streaming over the internet. *IEEE Computer Society*, pages 62–67, 2011.

[75] S. Soro and W. Heinzelman. Camera selection in visual sensor networks. In *Proc of the Advanced Video and Signal Based Surveillance (AVSS '07). IEEE Conference on*, pages 81–86, 2007.

[76] S. Soro and W. Heinzelman. A survey of visual sensor networks. *Advances in Multimedia*, pages 1–22, 2009.

[77] T. Stockhammer. Dynamic adaptive streaming over http standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems (MMSys11)*, Feb. 2011.

[78] Tiia Sutinen, Janne Vehkaperä, Esa Piri, and Mikko Uitto. Towards ubiquitous video services through scalable video coding and cross-layer optimization. *EURASIP Journal on Wireless Communications and Networking*, pages 1–17, 2011.

[79] A. Terras. *Fourier Analysis on Finite Groups and Applications*. Cambridge University Press, 1999.

[80] S.G. Wilson. *Digital Modulation and Coding*. Prentice Hall, 1995.

[81] W. Wolf, B. Ozer, and T. Lv. Smart cameras as embedded systems. *Computer*, 35:48–53, 2002.

[82] D. Yang, J. Shin, A. O. Ercan, and L. Guibas. Sensor tasking for occupancy reasoning in a network of cameras. In *Proc. of the IEEE/ICST 1st Workshop on Broadband Advanced Sensor Networks (BASENETS '04)*, San JosÃ©, CA, USA, 2004.

[83] Z. Yang and K. Nahrstedt. A bandwidth management framework for wireless camera array. In *Proc. of the international workshop on Network and operating systems support for digital audio and video (NOSSDAV '05)*, pages 147–152, Stevenson, Washington, USA, 2005.

[84] N.H. Zamora and R. Marculescu. Coordinated distributed power management with video sensor networks: Analysis, simulation, and prototyping. In *Proc. of the Distributed Smart Cameras (ICDSC '07). 1th ACM/IEEE International Conference on*, pages 4–11, 2007.

[85] Fan Zhang and H.D. Pfister. Analysis of verification-based decoding on the q-ary symmetric channel for large q. *IEEE Trans. Inf. Theory*, 57(10):6754–6770, October 2011.

[86] Y. Zhang and William E. Ryan. Structured ira codes. *IEEE Transactions on Communications*, 2007.

[87] N. Zhou, X. Zhu, Y. Huang, and H. Lin. Novel batch dependant cross-layer scheduling for multiuser OFDM systems. In *IEEE International Conference on Communications*, pages 3878–3882, Beijing, May. 2008.