MASTER THESIS

TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

TITLE: Cross-layer optimizations for multimedia services over mobile networks

TITULATION: Ingeniero Superior de Telecomunicación

AUTHOR: José Luis Asensio Aragón

DIRECTOR: Markus Rupp

SUPERVISOR: Luca Superiori, Martin Wrulich

DATE: 30-1-2010

To the clouds, the rain and finally the sun, a good harvest.

Overview

The objective of this thesis is to design a scheduler based on the cross layer concept, which improve the video transmission quality in HSDPA networks. This scheduler will be compared with the actual implementation to assess the achievable gain. The comparison will be conducted in a HSDPA system-level simulator to obtain performance results.

The proposed improvements will mainly be based in a modification of the video coding procedure and the HSDPA Node B scheduler. The encoder will be capable of signalizing the importance of the type of video frame and the scheduler will use this information to efficiently transmit to a user with good channel quality and considers fairness constrains.

In the first two chapters, I will resume the basic concepts of video encoding and HSDPA networks. In the video-coding part, I will talk about the standard H.264/AVC, its applications, its profiles and the images types. In the consecutive section, I will then explain the main functions and the important aspects of HSDPA technology.

Afterwards, a definition of cross layer design is given and I will also give an explanation of ways to improve the video transmission. The fourth chapter then goes deeply into the video coding modifications to obtain better decoding results. All investigated scheduler types are explained in the fifth chapter. The next chapter corresponding explains the proposed proportional fair scheduler with content awareness and the HARQ mechanism implemented.

The seventh chapter shows some simulation results, comparing the different combinations of coding characteristics and schedulers.

Finally, the conclusions will be discussed.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Traffic on future wireless networks is expected to be a mix of real-time traffic such as voice, multimedia teleconferencing and games, and data-traffic such as WWW browsing, messaging and file transfers. All of these applications will require widely varying and very diverse quality of service (QoS) guarantees for the different types of offered traffic. Various mechanisms have been proposed and since 1999 deployed to support data traffic over wireless channels to satisfy the increasing data-rate demands.

In this document, I address the issue of cross-layer networking, where the physical and MAC layer knowledge of the wireless medium is shared with higher layers, in order to provide efficient methods of allocating network resources and for data transmissions. In essence, future networks will have to adapt to the instantaneous radio channel conditions and capacity needs with the traffic and congestion conditions found in the packet-based world of the Internet. Further, such adaptation will need to be coordinated with a wide range of particular applications and user expectations, making the topic of cross-layer networking an increasingly important one for the evolving wireless access.

The Internet today consists of thousands of access networks, which vary in scale from large wire line networks, supporting thousands of users, to smaller wireless networks supporting hundreds of users. All these networks are interconnected by core networks which support millions of users. Mainly core networks offer two types of services: guaranteed service and best effort service. The first one provides some sort of service guarantee to individual users or groups of users. The other service makes no promises about the traffic to transmit, but is designed to maximize the throughput.

In mobile networks the service provided by the vendor should match as closely as possible the expectation of the end user in terms of objective parameters, such as delay or bandwidth. Thinking of media applications, such as gaming or video streaming; however some particular packets have more impact on the perceived end user quality. This motivates the need to distinguish the susceptible packets within an application data flow.

In this thesis, I have investigated how a cross layer approach can be implemented in High Speed Download Packet Access (HSDPA) [9] networks, treating different kinds of packets according to their priority. With video streaming as the study case, a cross layer approach to increase the QoE is proposed. This approach relies on the separation of the connection between the application server and the end user into multiple logical paths, giving an appropriate QoS class to each logical connection [13]. Thus, each packet can be mapped onto a path, depending on its relative importance for the end user.

The proposed cross-layer design will be simulated using a HSDPA system level simulator [10] [11], which will serve to compare the performance of different common scheduling types, such as Round Robin and PF variations. The results are presented in terms of the luminance Peak Signal to Noise Ratio (Y-PSNR), a distortion metric correlated with the quality of the video as perceived by the users.

This work is structured as follows: chapter I offers an overview of the HSDPA network. Chapter II introduces necessary video coding and transmission concepts. In Chapter III a definition of the concept cross layer is given, and in addition the design strategy to cover the video streaming requirements. The modifications done in the video coding and an explanation of the common schedulers are discussed in Chapter IV and V, respectively. In chapter VI, I explain the necessary HSDPA changes to achieve the concept of cross layer. The simulation results and the conclusions drawn in Chapter VII and VIII conclude the document.

1. CHAPTER: H.264/AVC OVERVIEW

H.264/AVC is one of the latest video coding standards, based on hybrid block video compression. The development of this codec is a responsibility of the Moving Pictures Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). The given name, H.264/AVC, is a mix between the names of both organizations standards, Advanced Video Coding for the MPEG4 and H.264 for the ITU-T.

The standard objectives are to improve compression and streaming capabilities. The two targets try to form an unique and a simple video coding design, which provides a ”network-friendly” video representation to storage, broadcast, or stream videos. Also new services and the increasing popularity of high definition TV have been considered to obtain a higher coding efficiency, high quality and low bitrate.

In the past, both standardization groups had focused only on their own standard. The most important design goal to being improved by MPEG was the video storage. On the other hand, VCEG focused on obtaining a better video streaming. Below, the developed standards of each organization are illustrated.

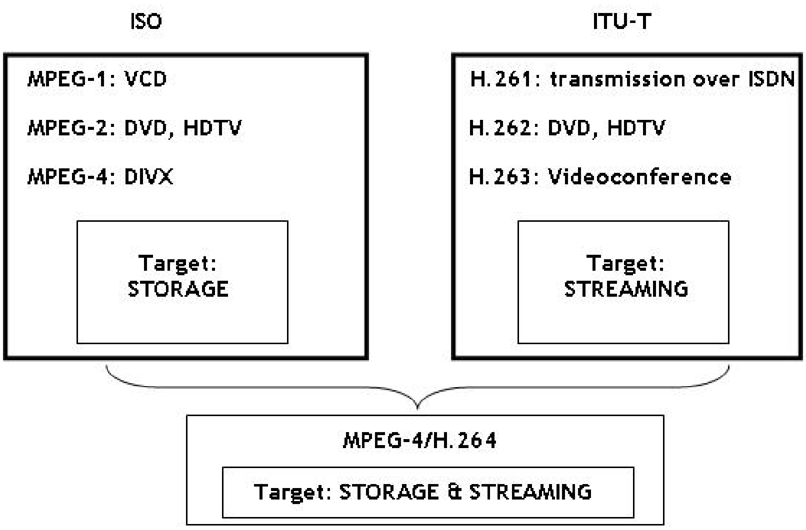


Illustration 1: Standardization Scheme.

The starting point of the video and audio compression standards of the ISO Moving Pictures Group was the MPEG-1 standard, introduced in 1991. This codec is used by Video CD (VCD) and includes the popular format of audio compression MP3. The standard MPEG-2, is an extension focused on the generic encoding of moving pictures and audio information. It is generally used for the video and audio compression, which includes: terrestrial TV (DVB-T), satellite TV (DVB-S), cable TV (DVB-C), High Definition TV (HDTV) and, it is also used by SVCD (Super VCD) and DVD (Digital Versatile Disc). After this second standard, ISO MPEG released the standard MPEG-3. MPEG-3 was designed to be a video compression standard for the High Definition TV (HDTV), but

MPEG-2 could be improved to achieve similar results with this earlier standard; thus MPEG-3 was not further developed.

Similarly, the ITU-T Video Experts Group developed its first video standard, the H.261, in 1990. This standard originally was designed for transmission over ISDN lines, supporting two image resolutions: QCIF (Quarter Common Interchange Format) which is 144x176 pixels and CIF (Common Interchange Format) which is 288x352 pixels. Then they designed the H.262 standard, which is very similar to the MPEG-2, targeting for DVD and HDTV. The reason of this development was the joint work between the ITU-T and the ISO organizations. In 1996, ITU-T presented the new standard H.263 which is a low-bitrate compressed format standard for videoconferencing.

In 2003 both organizations presented a new video compression standard, the H.264, under the name of Joint Video Team (JVT). The standard name follows the ITU-T naming convention, while the MPEG-4 AVC name relates to the naming convention in ISO/IEC MPEG. The objectives of the organizations were to develop a new standard without increasing the complexity and making it more compatible with much more applications than the previous standards. The basic functional elements are only slightly different to difference from the previous standards. Important changes in H.264/AVC occur in the details of each functional element [7].

1.1. Applications

The main purpose of H.264/AVC is to obtain maximum compatibility with most of the existing applications that are based on the previous standards. The following list enumerates the most important applications defined in Draft ITU-T Recommendations [6]:

1. Cable TV on optical networks (CATV).
2. Direct broadcast satellite video services (DBS).
3. Digital subscriber line video services (DSL).
4. Digital terrestrial television broadcasting (DTTB).
5. Interactive Storage Media (ISM): optical disks, etc.
6. Multimedia Mailing (MMM).
7. Multimedia services over packet networks(MSPN).
8. Real-time conversational services (RTC): videoconferences, videophone, etc.
9. Remote video surveillance (RVS).
10. Serial Storage Media (SSM).

In the era of media services, a compatible standard like H.264/AVC has opened the doors to new opportunities. In fact, many companies have begun to develop of new applications utilizing the freedom granted by the standard. An example is ”mobile TV”, the reception of audio-visual content on cell phones or portable devices. Several such systems for mobile broadcasting are currently considered for commercial deployment:

- Digital Multimedia Broadcasting (DMB) in South Korea.

- Digital Video Broadcasting - Handheld (DVB-H) in Europe and United States of America.
- Multimedia Broadcasting/Multimedia Service (MBMS) as specified in Release 6 of 3GPP.

In these three mobile TV services, the goal is to obtain a better video compression with H.264/AVC. Another field of application is for satellite TV services. Important enterprises and companies of satellite TV distributors have announced deployments of H.264/AVC.

1.2. H.264/AVC in wireless environments

H.264/AVC is the best standard so far for wireless systems, due to good behavior in video compression efficiency and error resilience. The main limitation of the wireless environments is the relatively low data-rate of the radio-link, which requires an efficient video coding standard. But not all depends on the bit rate in a video transmission also other random parameters change the conditions of the channel. Accordingly, the designers must use an appropriate receive buffer and initial delay to achieve a good service quality.

Another advantage of H.264/AVC is the network-friendly design. This is achieved by the separation into two layers, namely the coding action and the output fase. The layer that is in charge of the coded video transportation is called Network Abstraction Layer. And the layer that supports the encoding is named Video Coding Layer.

The Network Abstraction Layer (NAL) is a huge step supported in communications protocols, because it adapts the format of the video to the transmission. It also provides adapted header information for different transport layers or storage media [8]. The next figure depicts the standard in transport environment, showing the isolation made by the NAL between the video coder/decoder functionalities and the different lower transport layers which can coexist.

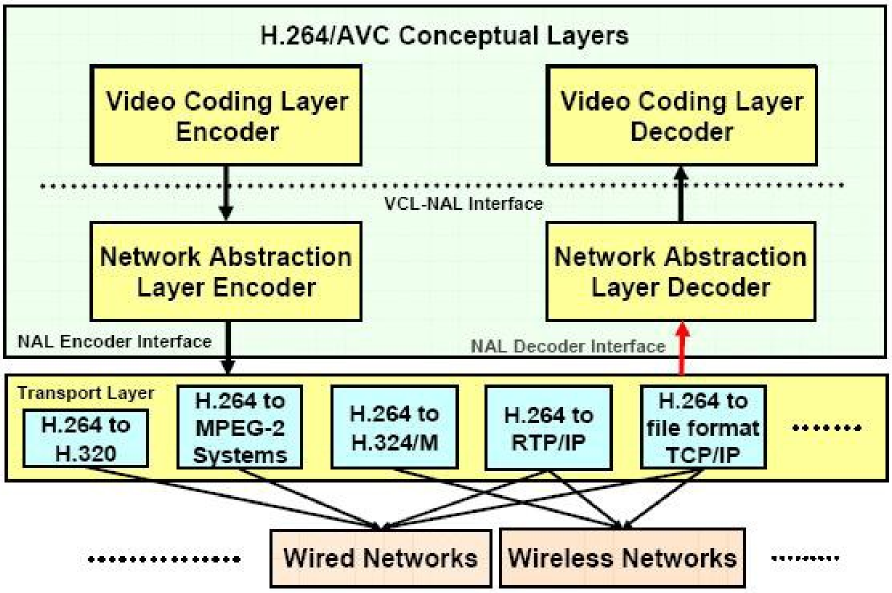


Illustration 2: H.264/AVC standard in transport environment [8].

1.2.1. NAL Unit

The NAL encapsulates the data from the VCL to enable transmission over packet networks or multiplexed environments. Data such as picture slices and parameter sets are sent from the VCL to the NAL and encapsulated into so called NAL units. These NAL units are used in the transport layer mapping. This structure of H.264/AVC allows a flexible operation over a variety of network environments [5].

The format of a NAL unit is shown in Illustration 3. A NAL unit consists of a 1-byte NAL header and a variable byte length payload. Data such as picture slices (coded video data) and parameter sets are stored in the payload field. The NAL header consists of one forbidden bit, two bits NRI (NAL Reference Index) whether or not the NAL unit is used for prediction, and five bits NUT (NAL Unit Type) indicating the type of the NAL unit. There are also some payload trailing bits used to adjust the payload to become a multiple of bytes. The trailing bits start with a “1” and are followed by multiple “0s”.

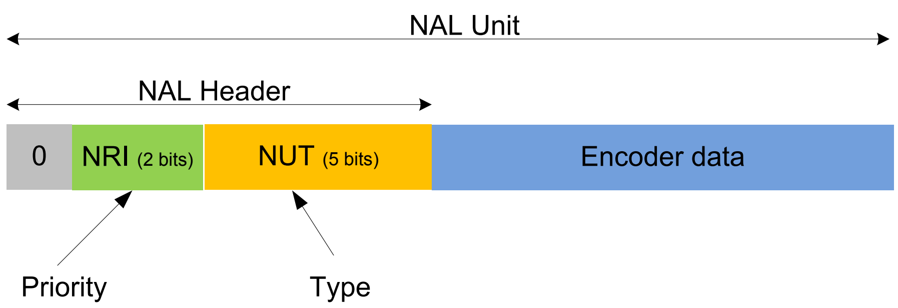


Illustration 3: NAL Unit Fields.

There are actually 12 different types of NAL units which are listed in table below. NAL unit type 1-5, and 12 are coded video data called VCL NAL units. The rest of the NAL unit types are called non-VCL NAL units and contain information such as parameter sets and supplemental enhancement information. Of these NAL units, IDR Pictures, SPS, and PPS are of special importance.

Type	Name
0	[Unspecified]
1	Coded Slice
2	Data Partition A
3	Data Partition B
4	Data Partition C
5	IDR (Instantaneous Decoding Refresh) Picture
6	SEI (Supplemental Enhancement Information)
7	SPS (Sequence Parameter Set)
8	PPS (Picture Parameter Set)
9	Access Unit Delimiter
10	EoS (End of Sequence)
11	EoS (End of Stream)
12	Filler Data

13-23	[Extended]
24-31	[Undefined]

Table 1:NAL types

An instantaneous decoding refresh (IDR) picture refreshes all the information of the video sequence. When the decoder receives an IDR picture, it behaves as if new coded video sequence begins. Therefore, pictures prior to this IDR picture are not considered.

A sequence parameter set (SPS) contains important header information that applies to all NAL units in the coded video sequence. A picture parameter set (PPS) contains header information that applies to the decoding of one or more pictures within the coded video sequence.

The transmission order of parameter sets and slices is restricted; that is, a parameter set must be sent to the decoder before the slice data that refer to that parameter set arrives at the decoder.

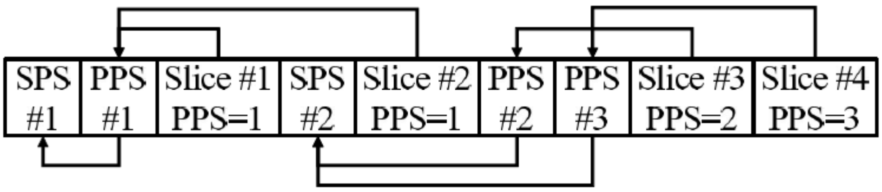


Illustration 4: Relationship between Parameter Sets and Slices.

The NAL Reference Index indicates whether a NAL is used for prediction or not. It can be useful when an application want to mark the importance of a NAL. This feature will be exploited by the cross-layer approach.

1.3. Profiles

The interoperability between encoder and decoder implementations is specified by the profiles, establishing a connection for various applications that have similar functional requirements.

Six profiles are actually defined in the standard, but only three are usually used: baseline (BP), main (MP) and extended (EP). The remaning profiles, high, high 10 and high 4:2:2, have their own characteristics, focused on particular applications. Illustration 3 shows the features of each one.

For all profiles a set of coding tools or algorithms are defined that are used to generate a compliant bit stream. On the receiver side, all bit streams must be decodable. Accordingly, decoders have to support all features for a specific profile, but encoders are not required to make use of any particular set of features supported in a profile, they only have to provide conforming bit streams.

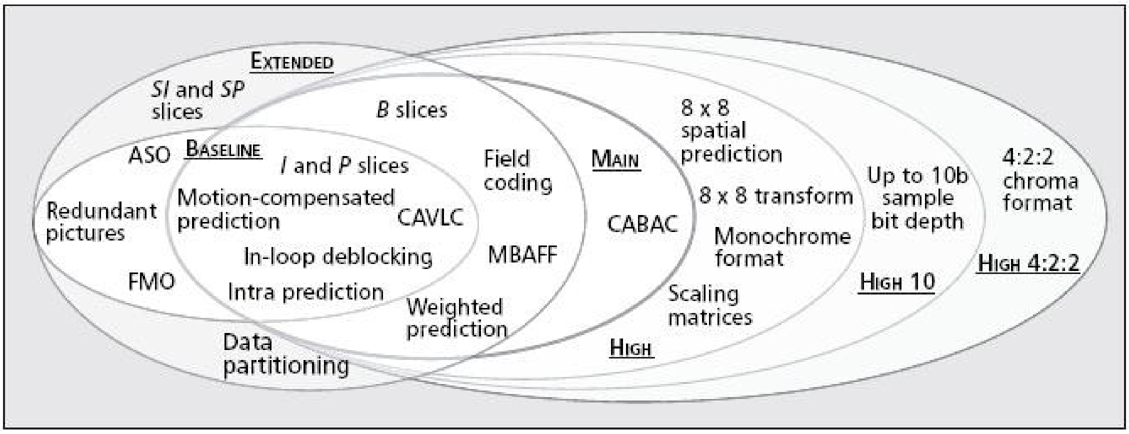


Illustration 5: H.264/AVC Profiles [5].

1.3.1. Baseline

This is the simplest profile and includes the basic features of the H.264/AVC standard:

- Only I and P slice types may be present.
- Flexible Macroblock Ordering (FMO).
- Arbitrary Slice Ordering (ASO).
- Redundant Pictures.
- Motion-compensated prediction.
- In-loop deblocking.
- Intra-prediction.
- Context Adaptative Variable Length Coding (CAVLC).

This profile emphasizes coding efficiencies and robustness with low computational complexity, and is thus ideal for embedded systems.

The features not supported by the baseline profile are:

- B slices.
- Weighted prediction.
- Picture or macroblock adaptive switching between frame and field coding.
- SP and SI slices

1.3.2. Main

The Main profile also emphasizes the coding efficiency, but typically provides a higher quality than the baseline, due to the B-slices and CABAC (Coding Adaptive Binary Arithmetic Coding). The performance is more complex and takes more coding time. This second profile includes all the features of the baseline but without FMO, ASO and Redundant Slices.

It also includes:

- B slices.
- Field coding.
- Weighted prediction.
- Macroblock adaptive frame-field (MBAFF).
- CABAC.

Only a subset of the coded video sequences that are decodable by a Baseline profile decoder can be decoded by a Main profile decoder.

1.3.3. Extended

The Extended profile is focused on obtain high robustness and flexibility, but without losing coding efficiency. This profile can be considered to be a superset of the Baseline and Main profiles, because it supports all tools in the specifications except CABAC. Another important point is that SP/SI slices and slice data partitioning tools are included in this profile.

1.4. Frames types and format

Usually the existing video coding standards, MPEG-2, H.263 and MPEG-4, define three main types of frames: I, P and B slices. H.264/AVC supports these three types and adds two new types: SP and SI slices.

A frame is a picture of a video sequence; each picture is divided into macroblocks which is the smallest unit of a video picture that covers a rectangular area taking 16x16 samples of the luma component (brightness) and 8x8 samples of each of the two chroma components. These samples serve to code the video sequence.

The macroblocks are organized in slices, which represent regions of a given picture that can be decoded independently of each other. Illustration 6 depicts an example.

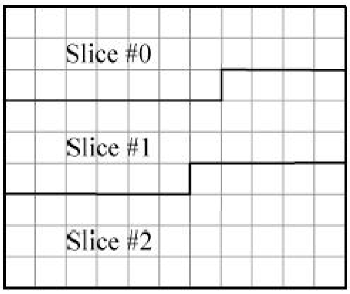


Illustration 6: Subdivision of a frame into slices.

1.4.1. Frame types definition

- <u>**I or "Intra" slices:**</u> this frame-type coded using Intra prediction and thus do not refer to any previous slice of the video sequence. They only contain reference from themselves. Always the first frame of a sequence is of this type, when no reference is given. Accordingly all profiles support them.

- **P or ”Predicted” slices:** these frames are coded using Inter prediction utilizing a prediction model from one or more previously encoded video frames. The model is formed by shifting samples in the reference frame(s) (motion compensated prediction). The AVC CODEC uses block-based motion compensation, the same principle adopted by every major coding standard since H.261. With at least one motion compensated prediction signal per prediction block. All profiles support this frame-type.

- **B or ”Bi-predicted” slices:** these frames are coded using Inter prediction with two motion-compensated prediction signals per prediction block that are combined using a weighted average. Baseline profile doesn’t support this type.

- **SP or ”Switching P” slices:** permit an efficient switching between two different bit streams coded at different bitrates, without the large numbers of bits required by I frames. They are only supported by the Extended profile.

- **SI or ”Switching I” slices**: are encoded using Intra prediction, allowing an exact match with SP slices for random access or error recovery. They are only supported by the Extended profile.

2. CHAPTER: HSDPA OVERVIEW

In this chapter, I will give a brief description of the mobile broadband technology for which the cross-layer optimizations are developed. This technology is called High Speed Packet Access (HSPA) and formed by HSDPA and its uplink pendant High Speed Uplink Packet Access (HSUPA). HSPA is the natural evolution of WCDMA networks and its main objective is to increase the spectral efficiency in down- and uplink data channels.

The target of HSDPA is to increase the peak data rates (current HSDPA deployments support downlink speeds of 1.8, 3.6, 7.2 and 14.4 Mbps), improve the quality of service, and enhance the spectral efficiency for burst packet data services. Also, HSDPA was designed to co-exist with R'99 in the same frequency band of 5 MHz.

Although having to share the data rate among the users located in a sector, HSDPA is able to satisfy the most common and demanded multimedia services such as email attachments, PowerPoint presentations or web pages.

2.1. Standardization

At the end of 1998, USA, Europe, Korea and Japan set up a forum named the 3rd Generation Partnership Project (3GPP). The objective of this joint venture is to introduce a new single global standard for mobile communication.

The first work on WCDMA was published in the Release 99 at the end of 1999. This document gave the first full series of WCDMA specifications. Two years later, Release 4 specifications were issued. In the meantime of the publication of Release 4 it became obvious that some improvement for packet access would be needed [9].

In March 2000, the feasibility study for HSDPA was started. The study was initially supported by Motorola and Nokia from the vendor side and BT/Cellnet, T-Mobile and NTT DoCoMo from the operator side. These companies wanted to give some improvements to be done over Release 99 specification. The main topics included in the further study were:

- Physical layer retransmissions.
- BTS-based scheduling.
- Adaptive coding and modulation.
- Multi-antenna transmission and reception technology, called 'multiple inputs multiple outputs' (MIMO).
- Fast cell selection.

The performed study gave some feasible improvements, which could be reached by the introduction of some of the studied techniques. Thus, HSDPA specifications were published in Release 5 in March 2002. At the beginning MIMO was not included in the specifications, but later it was added in Release 7. Fast cell selection, was discarded by the feasibility study, concluding that the complexity introduced would not justify the benefits [9]. The next picture shows the timeline standardization followed.

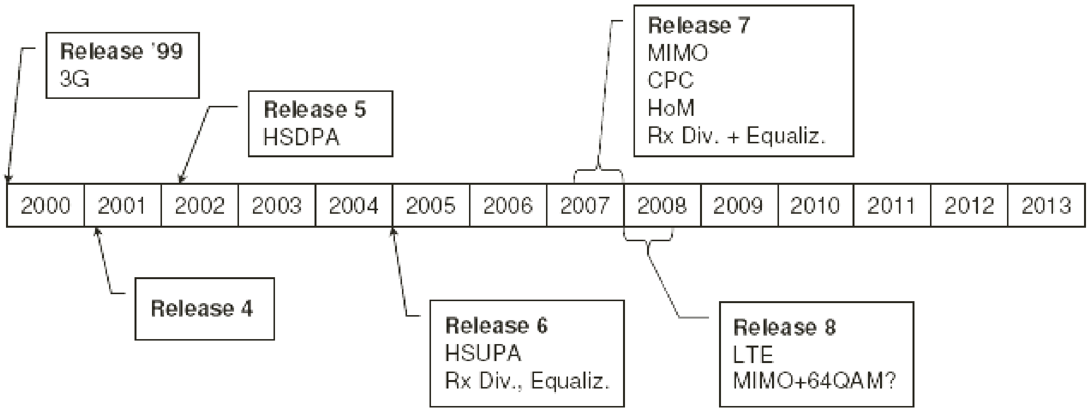


Illustration 7: 3GPP release timeline [9]

2.2. HSPA architecture and protocols

This section covers the impact of the HSDPA and HSUPA on the radio network and protocol architecture as well as on the network element functionalities and interfaces.

2.2.1. Radio resource management architecture.

The Radio Resource Management (RRM) functionality of HSPA has experienced changes compared to Release 99. One of the most prominent changes has been the shift of the scheduling function from the RNC to the Node B, and the according change of the RRM architecture. Consequently Node B has acquired more tasks and complexity. The base station has to take care of the QoS requirements, the buffer status for both interfaces (Iub, Uu) and the received channel feedback; but in fact all of these aspects are connected by the scheduling function.

The RNC is still the final entity responsible for the QoS parameters, MAC-d retransmission issues and flow control. Other functions are also maintained in the RNC, like the initial power set up and the admission control of the mobile network, as well as the handover processes.

User equipments now have more importance role in the network, because they report a parameter (CQI) related to the channel quality. This action provides the necessary information to the Node B to adapt the number of spreading codes, coding rates, and modulation to be used in next transmissions. This channel quality parameter can also be used in the scheduler.

The addition of the ARQ function has added a new function to the Node B and to the Terminal equipments. Both network elements now have to coordinate the transport block retransmission. The HSDPA functionality looks like depicted in illustration 8 [9].

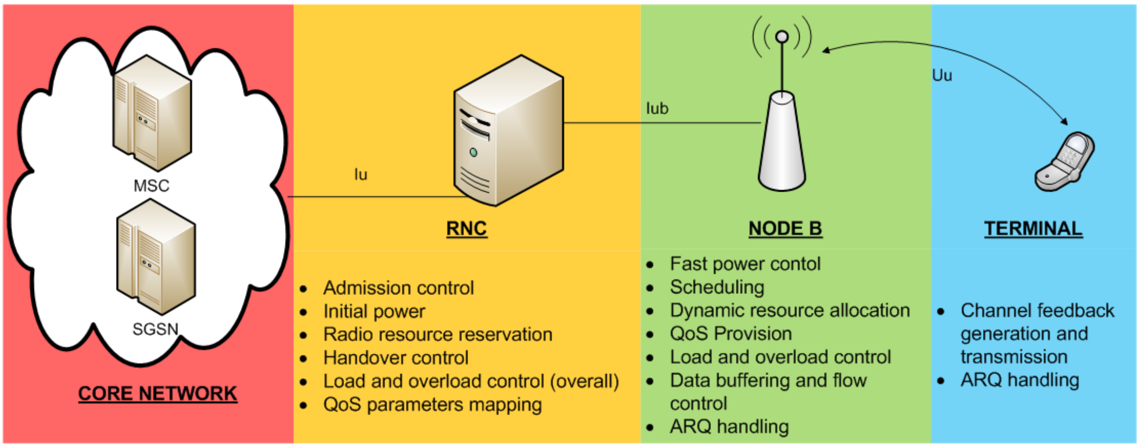


Illustration 8: HSPA RRM architecture.

The scheduling functionality of the Node B gives to the network more safety, when a user has to be reached. Because the Node B now knows the channel parameters of the users, as it can be shown in the next picture. Thus, by using these transmission data, the scheduler can be more efficient and select the user with the best channel conditions or be aware of the data content to send.

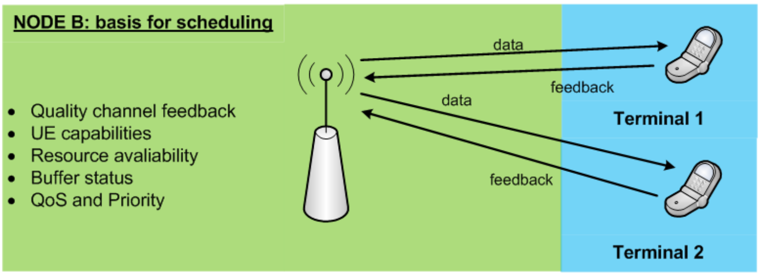


Illustration 9: HSDPA Node B scheduling principle.

2.2.2. HSDPA and HSUPA user plane protocol architecture

Now, as I was mentioned in the last section, the Node B has acquired more functions. The new functions of the Node B are included in a new MAC entity called MAC-hs, that is in charge of: handle the HARQ functionality of every HSDPA user, distribute the HS-DSCH resources between all the MAC-d flows according to their priority (i.e. Packet Scheduling), and select the appropriate transport format for every TTI (i.e. link adaptation).

The radio interface layers are not modified compared to the Release 99 architecture because HSDPA is intended for the transport of the existing logical channels. The MAC-hs also stores user data to be transmitted across the air interface, which imposes some constraints on the minimum buffering capabilities of the Node B. The move of the data queues to the Node-B creates the need of a flow control mechanism (HS-DSCH Frame Protocol) that aims at keeping the buffers full. The HS-DSCH FP handles the data transport from the serving RNC to the controlling RNC (if the Iur interface is involved) and between the controlling RNC and the Node-B, [9].

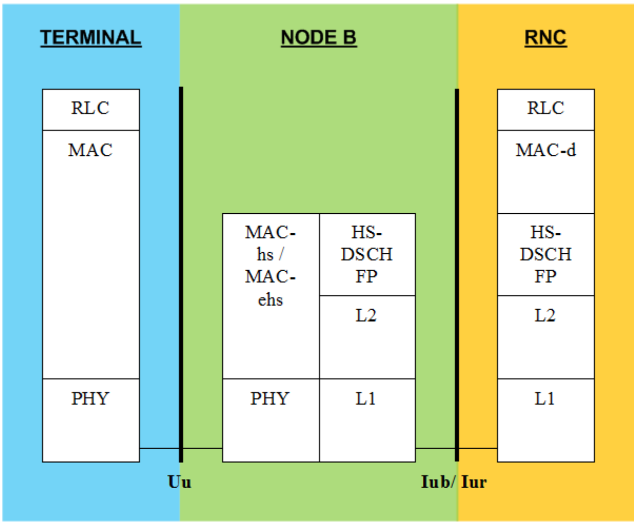


Illustration 10: HSPA protocol architecture [13].

The RLC layer is in charge of separating the lower layers of the protocol stack from upper layers. The available transfer modes are [13]:

- **TMD (Transparent Mode Data):** No overhead is added in this mode to the RLC SDU.
- **UMD (Unacknowledged Mode Data):** A sequent number is introduced in the RLC SDU in this mode, but the SDUs are not acknowledged.
- **AMD (Acknowledged Mode Data):** This is the most reliable mode that the RLC layer has, and the SDUs are prepared with a sequent number and to allow retransmissions (on the MAC-d).

2.3. HSDPA principles

This section explains the principles of HSDPA for WCDMA, with a special interest on the relevant new features included in Release 5, 6 and 7 specifications. HSDPA has been designed to increase downlink packet data throughput of Release 99 due to the fast physical layer retransmission and transmission and the fast link adaptation controlled by the base station. First a comparison between Release 99 and HSDPA is made and then HSDPA key aspects are presented.

2.3.1. HSDPA vs. Release 99 DCH

In Release 99, there are three different channels specified for data packet transmission [9]:

- **FACH: Forward Access Channel.** This channel is used to transport small data volumes or connection set ups during state transfers. In HSDPA, it is used to carry signaling information when a terminal has changed its state. The secondary common control physical channel (S-CCPCH) is responsible to carry its signalization content, but FACH does not support fast power control, soft handover, or variable spreading code. If there is a need to carry mixed services, FACH cannot be used.

- **DCH: Dedicated CHannel.** The main channel of Release 99 and in Release 5 is always operated together with HSDPA. When circuit-switched service is demanded it runs always on the DCH. In Release 5 the uplink user data always goes through DCH, but in Release 6 there is an alternative, that is an enhanced version of DCH (E-DCH). DCH can carry any kind of service using a fixed spreading code and fixed allocation time, although these parameters can be changed from upper layers. The theoretical maximum peak rate is 2Mbps and retransmissions are coordinated in the RNC. It supports the use of soft handover, meaning that can be connected with more than one station at a time and receiving information from both of them. DCH allows also the use of fast power control feature.

- **Downlink shared channel (DSCH).** This one has been replaced in Release 5 by the new High-Speed Downlink Shared Channel (HS-DSCH) and therefore will be analyzed in furthers sections.

The following table compares DCH and HS-DSCH channels in some important features [9]:

Feature	DCH	HS-DSCH
Variable spreading factor	No	No
Fast power control	Yes	No
Adaptive modulation and coding	No	Yes
Multi-code operation	Yes	Yes, extended
Physical layer retransmissions	No	Yes
BTS-based scheduling and link adaptation	No	Yes

Table 2: DCH and HS-DSCH channel comparison.

2.3.2. High-speed downlink shared channel

The HS-DSCH is the transport channel which carries the current user data in HSDPA. In the physical layer, the HS-DSCH is mapped on the High-Speed Physical Downlink Channel (HS-PDSCH). The main differences to the Release 99 DCH-based packet data operation, which are described in [9], are as follows:

- Lack of fast power control. In HSDPA, link adaptation determines a suitable combination of codes, coding rates, and modulation.
- Support of higher order modulation than DCH. In Release 99 only 4QAM was available, but the last release standardizes 64QAM, 16QAM and 4QAM for the downlink.
- The scheduling is done by the base station every 2ms.
- The Physical layer is in charge of the retransmissions and retransmissions combining.
- Lack of soft handover. Data are sent from only one serving HS-DSCH cell.
- Lack of physical layer control information on the HS-PDSCH. This is carried on the HS-SCCH for HSDPA use.
- Multicode operation with a fixed spreading factor value of 16. Thus, UEs will be able to support up to 15 codes because other common channels require one of them.
- Only turbo-coding is used, with DCH also convolutional code could be used.

HS-DSCH coding

The use of turbo-codes outperforms convolutional codes, but it could be a limitation due to the slender code selection imposed.

The use of new modulation schemes has introduced some changes in the channel coding chain. Even a bit scrambling functionality has been introduced to avoid having long sequences of '1s' or '0s', therefore good signal properties for demodulation are ensured.

A hybrid-ARQ(HARQ) functionality has also inserted and consists of two-stage rate matching function. This new feature allows the redundancy tuning between versions of different retransmissions when the same data are transmitted. HARQ can operate in two modes, 'Chase combining' or 'incremental redundancy', depending on the matching rate [9].

- **In Chase combining,** the rate matching is identical between transmissions so the same bit sequence is sent. The receiver stores the received samples as soft values, and therefore the memory consumption is higher than if it was storing hard values.

- **Incremental redundancy** uses a different rate matching between retransmissions. Different code-rates are used in the retransmissions with the aim of having the correct data with all. This solution requires more memory in the receiver.

When the physical retransmissions fail, the next step of the retransmission is handled by the RNC like in Release 99.

HS-DSCH link adaptation

A 2ms TTI period gives to the system a large dynamics, in which the system adapts itself to the channel conditions. Thus, the base station decides each 2 ms which user is scheduled and also with which coding, modulation combination and transport block size.

Link adaptation is based on the CQI report of the UE. The dynamic range obtained using this technique can reach 30dB according to [9].

HS-DSCH modulation

In Release 99, the modulation for DCH channel was fixed at 4QAM. But with the successive releases, higher modulation orders were added to the standard up to 64QAM [9]. That means that with a higher modulation order a larger the number of bits that can be carried per symbol. But higher order modulations need additional decision boundaries and more signal quality because the symbols are too close in the constellation.

Multiple Input Multiple Output HSDPA

MIMO technology was introduced in Release 7,named HSPA+, it supports 2x2 downlink MIMO, which means the base station uses two antennas to transmit two orthogonal data streams to two receive antennas at the UE. The gain acquired with this schema is a throughput increase without having to increase the bandwidth or the transmitted power.

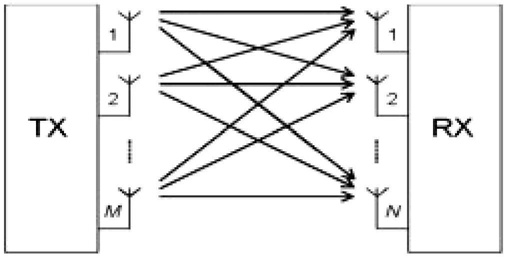


Illustration 11: MIMO schema.

MIMO needs a rich scattering environment to support the two data stream. Thus, MIMO benefits can be seen in dense urban areas where the size of the cells is typically small and there are many uncorrelated paths. In line-of-sight scenarios the MIMO gains are limited; however the fact of having a higher order modulation complements the MIMO results.

2.3.2. HSDPA physical layer operation procedure

The basic procedures of the HSDPA physical layer are described in this section [9].

- Each user feedback is evaluated every 2ms by the base station in order to select the user which will be scheduled. The selection criterion is not defined in the standard.
- Once a terminal has been determined as being served in a particular TTI, the base station identifies the HS-DSCH parameters needed for the transmission, including the number of codes, the modulation order and the UE capability limitations.
- The Node B starts to transmit the HS-SCCH two slots before the corresponding HS-DSCH TTI. This is because the first part of the HS-SCCH frame carries information needed to decode the HS-DSCH frame properly.
- The terminal monitors the HS-SCCHs (there can be up to four). Once part one of the HS-SCCH is decoded, it will start to decode part two and will buffer the necessary spreading codes from HS-DSCH.
- After decoding the HS-SCCHs parameters from part two, the UE can determinate to which ARQ process the data belong and whether there is the need of combining or not.
- The UE sends in the uplink direction an ACK/NACK indicator after the combination of the data (if applied) with a fixed delay, depending on the result of the CRC on HS-DSCH data.

2.3.3. HSDPA MAC layer operation

The HSDPA MAC layer has as key functionalities the scheduling in the Node B and the HARQ processes handling. The flow control features are also managed by the MAC layer, but usually are implemented together with the scheduling functionality. The specifications do not contain the implementation details of the scheduling schema which is left for individual network vendor decisions [9].

The MAC-hs PDU consists of a header and a payload which can carry more than one service data units (SDUs), also a potential padding field can be added to fit the available transport block size. The following fields are used [13]:

- **Version flag (VF):** to enable future protocol extensions.
- **Queue ID:** to allow different reordering queues at the terminal end. Note that only one transport channel may exist in a single TTI and in addition only one MAC-hs PDU.

- **Transmission sequence number (TSN):** parameter that facilitates the reordering operation at the UE.
- **Size index identifier (SID):** MAC-d PDU size.
- **N parameter:** indicates the number of MAC PDUs of the size indicated in the SID.
- **F field:** indicates whether there are additional SID and N fields in the MAC-hs header or not.

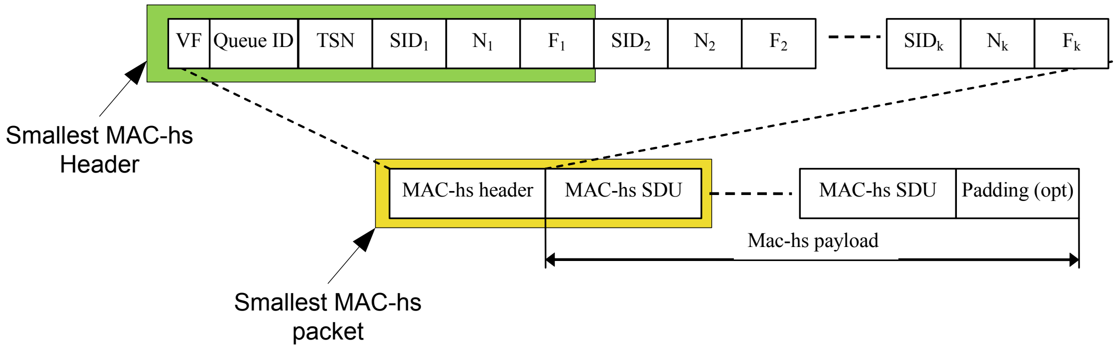


Illustration 12: MAC-hs PDU carrying several SDU´s[13].

The MAC layer one can multiplex different services into a single transport channel. This requires that both services should have similar QoS characteristics, as it is not possible for the Node B scheduler to separate them for different handling. In this case the multiplexing takes place in the RNC in the MAC-d layer, where several logical channels are multiplexed into a single MAC-d flow. A MAC-d header is also added to signalize the flow.

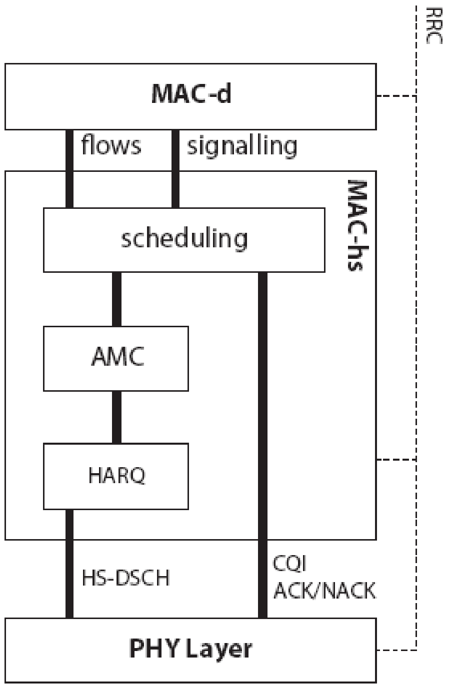


Illustration 13: Overview of the layer interaction responsible for the physical layer transmission in HSDPA [14].

To summarize, it has to be said that the link adaptation provides channel quality information to the Node B, thus allowing the MAC-hs scheduler to extract multiuser diversity within the cell. Data flow control is handled by the MAC-d, ensuring continuous data transfer even in case of handover. These mechanisms together make the physical layer of HSDPA very flexible and promising for exploiting cross-layer information.

MAC-d, MAC-hs and the physical layer set-up are controlled by the Radio Resource Control (RRC), which can adapt the network slowly to general performance targets. The available instantaneous Channel Quality Information (CQI) and the packet acknowledging (ACK/NACK) however are only available in the NodeB itself. The scheduler thus can act as an interface for the signaling information of upper layers and the CQI and ACK/NACK information of the wireless link. It has to be noted that in principle the scheduler may also influence the resources assignment of the individual users by affecting the modulation and coding settings as well as the retransmission handling [14].

3. CHAPTER: CROSS LAYER DESIGN

This chapter builds the motivation for cross layer improvements and explains the steps to build the connections among the HSDPA protocol layers are developed. First the term cross layer is defined, and then the requirements of the video processing are explored.

3.1. Definition and motivation

The basic idea of cross layer improvements is to dynamically transport feedback dynamically via layer boundaries to enable for compensation of e.g. overload, latency or other mismatch of requirements and resources [15]. This definition does not set up a new layer that works in parallel to the well known levels of a concrete communications protocol stack, but points the path to achieve a better quality of service with the interchange of communication parameters among layers. Also this definition contradicts the strict boundaries between layers established in the OSI communications model.

Nowadays, the different kinds of applications that use networks are very wide and all of them have different QoS requirements. Therefore the network behavior has to be adapted to these necessities, changing the scheduling mode or being aware of the data content. The scheduling mode is usually used to avoid congestion of a node or an empty buffer. The awareness of the data to transmit is used to prioritize some data flows over others that currently pass through the same network node.

In the case of the wireless networks, the case of study, the channel conditions affect the QoS dramatically, due to the random nature of the air propagation. Thus, achieving an acceptable QoS for a specific data flow is more complex, and needs more information interchange among the layers.

The second reason is the actual tendency to do everything wirelesses, aggravating the correct work of some common applications. Therefore communication protocols or applications must be adapted to the changing user requirements.

Focusing on the parameters where the layers can interchange, two distinct groups can be created. First, we find parameters referring to data information, denoting the importance and the characteristics of a data flow. This group can be provided by the upper layers. The second group is referred to feedback parameters that the user device sends to the network. This group is considered in wireless networks the most important because it can be used to report buffer states, channel conditions and acknowledges reports.

Schedulers usually are located in the MAC layer, where the parameters from the uppers layers and the feedbacked parameters can be converged. Thus, all this information can be used to schedule the user with the best criteria, as can be shown in the illustration 14.

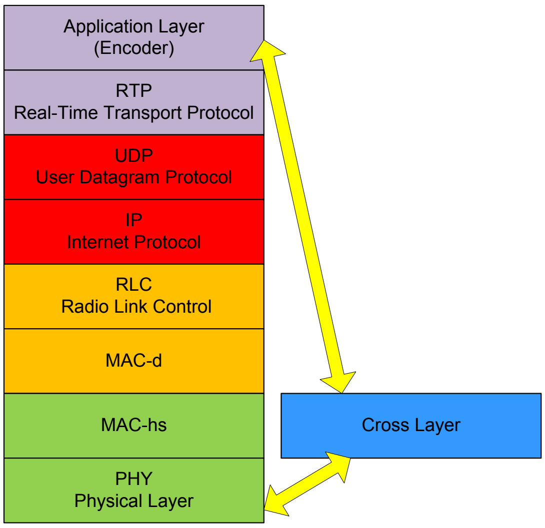


Illustration 14: Information interchanges between layers to create the cross layer optimizations.

3.2. Video steaming requirements

As a practical example, this thesis is focused on the video streaming transmission over an HSDPA wireless network. This technology is the door to the next mobile technologies.

Streaming can be more complex in a packet based network because they have strong and specific requirements. The video data flows must be formatted, denoting that the latency between consecutive packets must be the same, the data bit rate has to be high and constant, and the video packet loss rate must be close to zero.

Constant bit rate is needed to feed the decoder application in a proper way, and to see the video without interruptions. Almost all video applications have a predecoding buffer to stabilize the process, but these buffers usually have a finite size, which can lead to overflow problems. Therefore the transmission should be aware of the packet latency or jitter to avoid it.

However, the most desired case is to receive every video packet without errors to avoid decoding problems.

3.3. Design strategy

With the definition of cross layer interactions and the type of information that the layers have to interchange to achieve the video streaming transmission optimization (NAL priority, NAL types and channel parameters), an implementation strategy has to be presented. This path serves for the concept implementation, but also for the simulation tests.

The **First step** is to signalize the video frames to be easily recognized by the lower layers. Therefore marking the frames is needed to implement the cross layer improvements. This action takes place in the application layer, signalizing the priority of the different kind of frames. This step is explained in detail in the next chapter where the video modifications are explained.

The **Second step** in the strategy is to signalize the content of the PDUs through the protocol stack without changes on it. The target of this action can be achieved, fixing the HSDPA map for QoS class (application) into a DiffServ Class [17, 18].

The **Third step** is to design a content awareness scheduler, which uses the feedback and the data type information. This scheduler will have to consider the type of data, the status of the channel and the status of the data transmission for each user.

The proposed scheduler has to be designed to reach some basic specifications that one:

- Decrease the video packet error rate to almost zero.
- Maintain a constant bit rate to feed the decoder without overflows.
- Maintain the fairness among the different users who are served.
- Increase the video quality, measured in Luminance Peak Signal to Noise Ratio (Y-PSNR) for video.

In the practical example of H.264 video streaming, the NALs headers that contain the information about the type and the priority of the video frames have to be understood by the immediate lower layers. RTP layer (Real Transport Protocol) is the first that has to read these fields and mark the priority in its own headers fields, in the PT field (Payload Type) [19]. The PT field is able to signalize to the application layer.

Then the UDP protocol encapsulates the RTP SDUs containing the video NALs without taking care about what is on it. But the IP protocol with differentiated services activated is still able to signalize the types of data. To do that, it utilizes the differentiated services field of the IP header [17, 18]. At this point of the protocol stack a problem appears: the RNC (Radio Network Controller), which is in charge of managing the MAC layer, cannot read the IP headers. Therefore, a direct signalization of the priority and the type of data is not possible.

A possible solution for this problem is to divide the IP packets into priority data flows depending on the data types. These data flows are called PDP-Contexts [14]. These contexts with a specific QoS for its data type create a logical channel with the mobile terminal to send the data.

PDP-Contexts allow the system to specify independent QoS parameters for multiple applications running at the same mobile terminal. There are two different categories of PDP-contexts, named primary and secondary. Every mobile first has to open a primary context. Then the mobile can either attach a secondary to the primary or initiate further primary ones. Multiple primary PDP-contexts have different IP addresses and are typically used for different applications. Multiple secondary PDP-contexts share the same IP address and must be attached to a previous initialized primary context. Illustration 16 shows the setup differences.

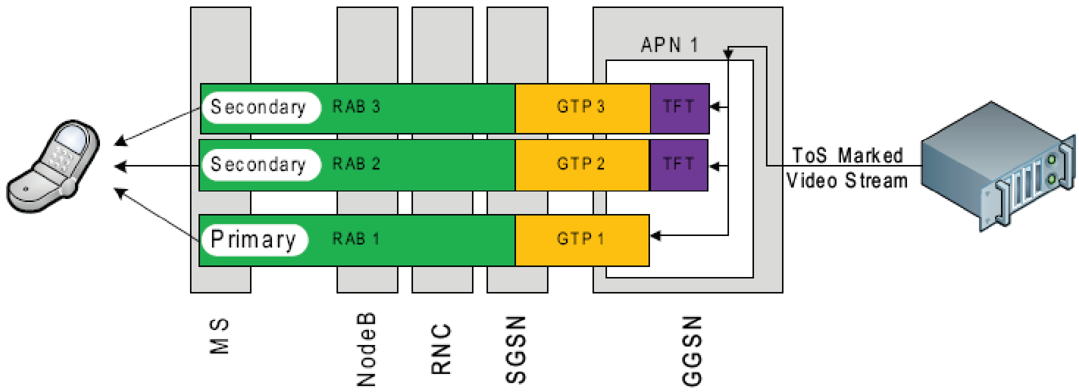


Illustration 15: PDP-Context Schema [14].

In our case for example, a primary PDP-context for the NALs with the higher priority, I frames, because they refresh the video sequence and are independent of the previous frames. The secondary PDP-context will carry the NALs of less importance. In order to split up the data into the different PDP-context for the same user, Traffic Flow Templates (TFT) are defined. A TFT can be seen as a packet filter applied onto each IP packet entering the GGSN. Also, there exists one TFT per secondary PDP-context. The filtering rules are based on one or more of the following attributes: source address, IP protocol number, destination port (range), source port (range), IPsec security parameter index and the type of service. The last attribute in this list makes the TFT the perfect match for the proposed method as it is the same header field used for DiffServ [14].

To sum up the encapsulation method used to transmit the video in our case of study, first the video NALs are encapsulated in a RTP PDU signalizing the type and priority in the header fields. Then it is encapsulated in a UDP datagram that later will be put into an IP packet marking the data content in the differentiated services. When the IP packets arrive at the GGNS with different QoS requirements, these are divided into different data flows called PDP-contexts according to different traffic flow templates. Afterwards, the IP PDUs are given to the MAC layer, where the scheduler will take the type of data flow to determine the QoS. If a user is selected to receive data, the IP PDU to send will be encapsulated in a transport block attaching a MAC Header and a CRC, as is shown in illustration 16. The size of each transport block depends on the channel quality according to the link adaptation procedure.

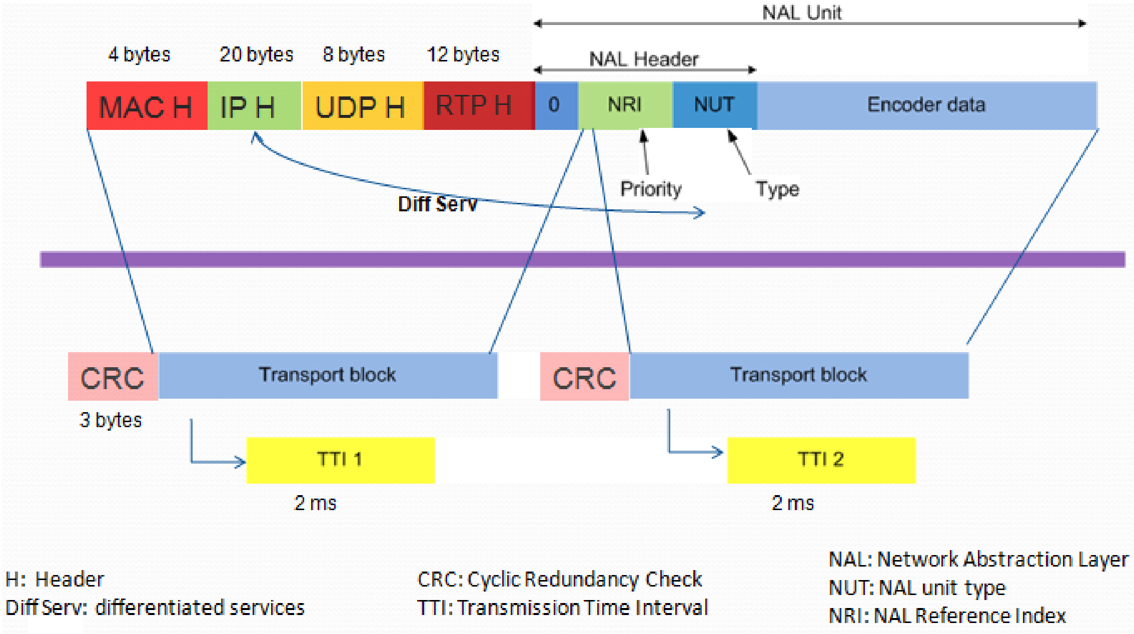


Illustration 16: Encapsulation method.

4. CHAPTER: VIDEO CODING MODIFICATIONS

In this chapter I will explain the modifications applied to the video encoder in order to mark the priority in the different NALs. Two parts have been modified, the decrease of the dependence among the video frames to send, and the default priority given to the NALs, these changes are explained respectively in the section 4.2 and 4.3. But in the first section I explain the coding settings.

4.1. Coding settings

The first thing to mention in this section is that the baseline profile of H.264/AVC is used, according to the 3GPP requirements. A cellular phone has limited power and it is necessary to take care of the batteries. Other profiles have better quality coding conditions, but they also need more computational requirements, more complexity, implying higher power consumption and are, therefore, not allowed by the 3GPP.

A resolution of 144x176 pixels, known as QCIF (Quarter Common Interchange Format), is chosen, to fit the receiver side screen. Encoder and decoder use a buffer size of one frame. One frame is coded and buffered to be the reference for the next temporally predicted frame.

As the baseline profile consist of I and P frames, it is necessary to define a suitable Intra period for the video transmission. This period is fixed to 45, which means one I frame for every 45 P. I frames refresh the reference of the video sequence, because are coded without any reference to previous ones. This kind of codification force to divide into slices of the I frame, due to the maximum payload assigned to the NALs, set at 750 bytes.

A frame rate is also defined to have a good relation between the subjective quality perceived by the user and the limited number of frames transmitted defined by the maximum bit rate of the network. The rate is fixed to 15 frames per second; one second of a video consists of 15 frames.

4.2. Non referenced P frames

The loss of a frame independently of the type, leads to a step backward in video quality. And this issue becomes worse, if an I frame is lost, because the reference for the next P sequence. But given a special priority treatment to the I frames, it will be solved. However, the loss of a P frame is also a big step backward, because all of them all have a temporal correlation. For this reason, a new type of P frame called non referenced P frame is created.

These new kind of frames try to decrease the dependency between P frames. Looking at the picture below, the common situation is to have one P frame referencing the previous frame. But with the non referenced P frames one frame is used to be the reference for the next two P frames to code, as is shown in the second example of the picture. A P frame is maintained in the coding buffer twice. Thus, one P frame is not used for the codification, the non referenced P frame. The loss of these frames is less important than the “normal” P frames, because they are not used for the decoding. Thus, the error propagation is allowed, the final objective.

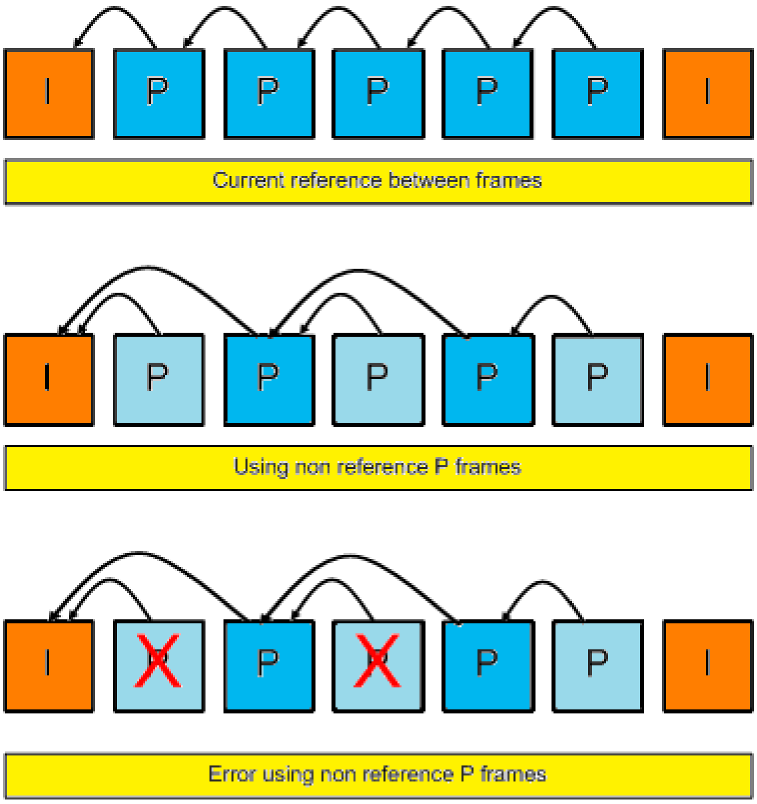


Illustration 17: Non referenced P Frames

To implement this new type of frames, the option DisposableP has been enabled in the codification settings. This field allows having the following encoding sequence: I0p1P2p3P4p5… where the numeric index corresponds to the frame index and display order, while uppercase and lowercase imply reference and non reference pictures respectively [4]. The next line is where the disposable P frames are set in the cfg file.

```
DisposableP             = 1  # Enable Disposable P slices in the primary layer
(0: disable/default, 1: enable)
```

It is necessary to modify the encoder to reach the objective of the non referenced P frame. The coder must maintain the same P frame in the buffer to code the following two P frames. This action is made by the next function, located in the coder file lencod.c.

```
/*!
 ***********************************************************************
 * \brief
 *    Initialize encoding parameters.
 ***********************************************************************
 */
```

```
void init_frame_params()
{
  int base_mul = 0;

  if (params->idr_period)
  {
    if (!params->adaptive_idr_period && ( img->frm_number - img->lastIDRnumber )
% params->idr_period == 0 )
      img->nal_reference_idc = NALU_PRIORITY_HIGHEST;

    if (params->adaptive_idr_period == 1 && ( img->frm_number - imax(img-
>lastIntraNumber, img->lastIDRnumber) ) % params->idr_period == 0 )
      img->nal_reference_idc = NALU_PRIORITY_HIGHEST;
    else
      img->nal_reference_idc = (params->DisposableP) ? (img->frm_number + 1)% 2
: NALU_PRIORITY_LOW;
  }
  else
    //img->nal_reference_idc = (img->frm_number && params->DisposableP) ? (img-
>frm_number + 1)% 2 : NALU_PRIORITY_LOW;
       img->nal_reference_idc = (((img->number)%2 == 0) && (img->type !=
I_SLICE)) ? 0 : NALU_PRIORITY_LOW;

  //much of this can go in init_frame() or init_field()?
  //poc for this frame or field
  if (params->idr_period)
  {
    if (!params->adaptive_idr_period)
      base_mul = ( img->frm_number - img->lastIDRnumber ) % params->idr_period;
    else if (params->adaptive_idr_period == 1)
      base_mul = (( img->frm_number - imax(img->lastIntraNumber, img-
>lastIDRnumber) ) % params->idr_period == 0) ? 0 : ( img->frm_number - img-
>lastIDRnumber );
  }
  else
    base_mul = ( img->frm_number - img->lastIDRnumber );

  if ((img->frm_number - img->lastIDRnumber) <= params->intra_delay)
  {
    base_mul = -base_mul;
  }
  else
  {
    base_mul -= ( base_mul ? params->intra_delay :  0);
  }

  img->toppoc = base_mul * (2 * (params->jumpd + 1));

  if ((params->PicInterlace==FRAME_CODING) && (params-
>MbInterlace==FRAME_CODING))
    img->bottompoc = img->toppoc;     //progressive
  else
    img->bottompoc = img->toppoc + 1;   //hard coded

  img->framepoc = imin (img->toppoc, img->bottompoc);

  //the following is sent in the slice header
  img->delta_pic_order_cnt[0] = 0;
```

```
  if ((params->BRefPictures == 1) && (img->frm_number))
  {
    img->delta_pic_order_cnt[0] = +2 * params->successive_Bframe;
  }
}
```

In the next two illustrations, you can see the results of the non referenced P frames implementation. The last column of the illustration 18 shows whether one frame has been used as a reference (value 1) or not (0). But in the illustration 19, there are ones and zeros interleaved, which means that the frames with the zero are not used as a reference, non referenced P frames.

Frame	Bit/pic	QP	SnrY	SnrU	SnrV	Time(ms)	MET(ms)	Frm/Fld	Ref
0000(NVB)	80								
0000(IDR)	16384	32	35.522	43.062	43.628	912	0	FRM	1
0001(P)	2104	30	35.987	43.287	44.194	2603	1315	FRM	1
0002(P)	2496	30	36.139	43.574	43.899	2618	1321	FRM	1
0003(P)	2448	30	36.286	43.658	44.303	2682	1366	FRM	1
0004(P)	2112	30	36.223	44.150	44.719	2697	1365	FRM	1
0005(P)	2288	30	36.145	44.242	44.439	2764	1436	FRM	1
0006(P)	2712	30	36.324	43.663	44.847	2793	1393	FRM	1
0007(P)	2016	30	36.162	43.847	44.364	2712	1367	FRM	1
0008(P)	2112	30	36.378	43.625	44.540	2674	1340	FRM	1
0009(P)	1976	30	36.585	43.868	45.115	2670	1359	FRM	1
0010(P)	3720	30	34.361	40.818	41.249	2780	1404	FRM	1
0011(P)	1728	30	35.078	42.112	42.736	2677	1364	FRM	1
0012(P)	2040	30	35.598	42.557	42.786	2704	1350	FRM	1
0013(P)	1048	30	35.600	42.260	42.499	2684	1347	FRM	1

Illustration 18: Coding Trace without Non Referenced P Frames.

Frame	Bit/pic	QP	SnrY	SnrU	SnrV	Time(ms)	MET(ms)	Frm/Fld	Ref
0000(NVB)	80								
0000(IDR)	16384	32	35.522	43.062	43.628	953	0	FRM	1
0001(P)	2104	30	35.987	43.287	44.194	2698	1383	FRM	1
0002(P)	2488	30	36.139	43.574	43.899	2536	1359	FRM	0
0003(P)	3256	30	35.855	43.270	43.476	2685	1350	FRM	1
0004(P)	2424	30	36.059	43.710	43.989	2516	1337	FRM	0
0005(P)	3216	30	35.716	43.611	44.263	2651	1326	FRM	1
0006(P)	2384	30	35.957	43.326	44.175	2504	1325	FRM	0
0007(P)	3088	30	35.741	43.109	43.189	2659	1332	FRM	1
0008(P)	2072	30	36.036	43.566	43.839	2506	1331	FRM	0
0009(P)	2800	30	35.934	43.307	43.902	2651	1331	FRM	1
0010(P)	4240	30	34.506	40.941	41.143	2560	1323	FRM	0
0011(P)	3952	30	34.880	41.714	41.995	2682	1329	FRM	1
0012(P)	2056	30	35.415	42.328	42.223	2506	1330	FRM	0
0013(P)	1960	30	35.304	41.698	42.264	2672	1344	FRM	1

Illustration 19: Coding Trace with Non Referenced P Frames.

Next two captures of video decoder trace show the quality loss without non referenced P frames and the gain achieved with this new type of frames. In the illustration 20, mobile network has lost the seventh NAL, which corresponds to the fourth frame which is a P frame. At the receiver's side, the decoder conceals the missing information, but the SnrY (instantaneous P-YSNR) decrease almost 4 dB units. SNRY serves to measure the difference in the luma component between the original frame and the decoded one.

Frame	POC	Pic#	QP	SnrY	SnrU	SnrV	Y:U:V	Time(ms)
00000(IDR)	0	0	32	35.5216	43.0625	43.6284	4:2:0	236
00001(P)	2	1	30	35.9869	43.2869	44.1936	4:2:0	92
00002(P)	4	2	30	36.1395	43.5738	43.8990	4:2:0	94
Warning: RTP sequence number discontinuity detected								
00003(P)	6	3	30	36.2861	43.6577	44.3035	4:2:0	97
00004(*P*)	8	4	30	32.9047	42.9985	44.0378	4:2:0	0
00005(P)	10	5	30	32.9495	42.9529	44.0231	4:2:0	94
00006(P)	12	6	30	33.1873	42.5519	44.4084	4:2:0	101
00007(P)	14	7	30	33.1295	42.7457	43.8948	4:2:0	93
00008(P)	16	8	30	33.2215	42.7358	44.0783	4:2:0	95
00009(P)	18	9	30	33.3670	43.2479	44.6215	4:2:0	85
00010(P)	20	10	30	32.2287	40.6198	41.0796	4:2:0	100
00011(P)	22	11	30	32.7722	41.9600	42.4497	4:2:0	85
00012(P)	24	12	30	32.9897	42.3708	42.5126	4:2:0	91
00013(P)	26	13	30	32.9676	42.0713	42.2359	4:2:0	77

Illustration 20: Decoding Trace with Quality Lost.

On the other hand, we have repeated the same test using non reference P frames. Accordingly, the mobile network has lost the same NAL which corresponds now to a non reference P frame. Then the decoder notices the lost, but it continues correctly decoding the sequence because no reference frame has been lost. Thus, SNRY maintains similar values and the decoder copy the previous frame in the space left by the lost frame.

Frame	POC	Pic#	QP	SnrY	SnrU	SnrV	Y:U:V	Time(ms)
00000(IDR)	0	0	32	35.5216	43.0625	43.6284	4:2:0	233
00001(P)	2	1	30	35.9869	43.2869	44.1936	4:2:0	89
00002(P)	4	2	30	36.1395	43.5738	43.8990	4:2:0	92
Warning: RTP sequence number discontinuity detected								
00003(P)	6	2	30	35.8549	43.2698	43.4761	4:2:0	102
00005(P)	10	3	30	35.7162	43.6107	44.2633	4:2:0	92
00006(P)	12	4	30	35.9575	43.3257	44.1750	4:2:0	86
00007(P)	14	4	30	35.7413	43.1094	43.1887	4:2:0	93
00008(P)	16	5	30	36.0355	43.5664	43.8391	4:2:0	84
00009(P)	18	5	30	35.9340	43.3071	43.9023	4:2:0	91
00010(P)	20	6	30	34.5062	40.9412	41.1429	4:2:0	103
00011(P)	22	6	30	34.8795	41.7138	41.9949	4:2:0	101
00012(P)	24	7	30	35.4147	42.3275	42.2225	4:2:0	84
00013(P)	26	7	30	35.3039	41.6979	42.2637	4:2:0	86

Illustration 21: Decoding Trace of a Video Sequence with Non Referenced P Frames without Quality Lost.

With the implementation of non reference P frames an advantage of 4 dB in the SnrY is achieved in the following frames of the sequence. If a mobile network is capable of signalizing this kind of frames, and sends them when the worst channel conditions are given, no temporal error propagation occurs when a non reference P frame is lost, because they are not so important for the decoding. So the next step is to signalize the frame type in the NAL header and in the rest of the headers of the protocol stack, building a differentiated service for each frame kind.

4.3. Priority implementation

H.264 codec allows the signalization of the different types of NAL units and their priority. The fields in the NAL header which cover such marking are the NAL unit type and the NAL reference index, respectively. In the original implementation of the encoder by looking only at the values of the NAL header we are not able to distinguish an I frame and a P frame. In both cases, NAL unit type is set as a *Coded slice* and the value priority *high* marked in the NAL reference index.

For this reason, we need to implement a new priority schema which gives more information about the frames carried by a NAL only looking at the header fields. The designed schema is shown in the following table.

Frames type	NAL unit type	Default Priority	New Priority
SPS	SPS	HIGHEST	HIGHEST
PPS	PPS	HIGHEST	HIGHEST
IDR	IDR	HIGHEST	HIGHEST
I	Coded slice	HIGH	HIGH
P	Coded slice	HIGH	LOW
B	Coded slice	NON USED	NON USED
SP	Coded slice	NON USED	NON USED
SI	Coded slice	NON USED	NON USED
P_NON-REFERENCED	Coded slice	NON USED	DISPOSABLE

Table 3:NAL priority schema.

Four priority signalization levels in the codec can be used to set the importance of the carried frame in the NAL headers. The more important NALs are those that carry the decoding settings, which are SPS, PPS and first I frame, called IDR, so they have the highest priority. In the next level, it is situated the I frames, because refresh periodically the sequence reference values. To distinguish the I and P frames in the NALs headers, the priority value given to the P frames was decreased, and the priority given to the P non referenced frames will be the lowest. The NAL types have not changed.

These changes have been made in the encoder application, as can be shown in the red lines of the *create_slice_nalus* function at the lencod.c file.

```
/*!
************************************************************************
* \brief
*    This creates a NAL unit structures for all data partition of the slice
*
************************************************************************
*/
void create_slice_nalus(Slice *currSlice)
{
  // KS: this is approx. max. allowed code picture size
  const int buffer_size = 500 + img->FrameSizeInMbs * (128 + 256 * img-
>bitdepth_luma + 512 * img->bitdepth_chroma);

  int part;
  NALU_t *nalu;

  for (part=0; part< currSlice->max_part_nr; part++)
  {
    if (currSlice->partArr[part].bitstream->write_flag)
    {
      nalu = AllocNALU(buffer_size);
      currSlice->partArr[part].nal_unit = nalu;
      nalu->startcodeprefix_len = 1+ (currSlice->start_mb_nr == 0 && part == 0
?ZEROBYTES_SHORTSTARTCODE+1:ZEROBYTES_SHORTSTARTCODE);
      nalu->forbidden_bit = 0;

      if (img->currentPicture->idr_flag)
```

```
      {
        nalu->nal_unit_type = NALU_TYPE_IDR;
        nalu->nal_reference_idc = NALU_PRIORITY_HIGHEST;
      }
      else
      {
        //different nal header for different partitions
        if(params->partition_mode == 0)
        {
          nalu->nal_unit_type = NALU_TYPE_SLICE;
        }
        else
        {
          nalu->nal_unit_type = (NaluType) (NALU_TYPE_DPA +  part);
        }

/****************************************************************************/

        if (img->nal_reference_idc !=0 && img->type == I_SLICE)
        {
                nalu->nal_reference_idc = NALU_PRIORITY_HIGH;
        }
        else if (img->nal_reference_idc !=0 && img->type == P_SLICE)
        {
                nalu->nal_reference_idc = NALU_PRIORITY_LOW;
        }
             else if (((img->frame_num)%2 == 0) || (img->type != I_SLICE ))
             {
                nalu->nal_reference_idc = 0;
             }
             else
             {
                nalu->nal_reference_idc = NALU_PRIORITY_DISPOSABLE;
             }
      }
    }
    else
    {
      currSlice->partArr[part].nal_unit = NULL;
    }
  }
}
```

In the table shown below, you can see the changes that have been made. The first column lists the NAL trace of a video with common coding, where no differences can be observed between I and P NAL headers, all have *Priority: 2* and *NAL unit type: coded_slice*. When the non referenced P frames are introduced, the priority schema changes a little given *Priority: 0 to* this type of frames, but yet is not possible to distinguish I and P frames.

In the last column, the new priority schema is implemented. It is possible to see at the beginning the SPS, PPS and IDR frames with the highest priority (3) then a combination of NALs with priority 0 or 1 (lowest and low) which are respectively non referenced P frames and “normal” P frames. At the bottom of the column, there is a NAL that carry an I frame, you can recognize it due to the high priority (2) and the bigger payload.

Values of NAL Header without any modification.	Values of NAL Header with Non referenced P Frames included.	Values of NAL Header with New Priority Implementation.
1) Priority:3 NAL unit type:SPS packet payload length: 7	1) Priority:3 NAL unit type:SPS packet payload length: 7	1) Priority:3 NAL unit type:SPS packet payload length: 7
2) Priority:3 NAL unit type:PPS packet payload length: 3	2) Priority:3 NAL unit type:PPS packet payload length: 3	2) Priority:3 NAL unit type:PPS packet payload length: 3
3) Priority:3 NAL unit type:IDR packet payload length: 2048	3) Priority:3 NAL unit type:IDR packet payload length: 2048	3) Priority:3 NAL unit type:IDR packet payload length: 2048
4) Priority:2 NAL unit type:coded_slice packet payload length: 263	4) Priority:2 NAL unit type:coded_slice packet payload length: 263	4) Priority:1 NAL unit type:coded_slice packet payload length: 263
5) Priority:2 NAL unit type:coded_slice packet payload length: 312	5) Priority:0 NAL unit type:coded_slice packet payload length: 311	5) Priority:0 NAL unit type:coded_slice packet payload length: 311
6) Priority:2 NAL unit type:coded_slice packet payload length: 306	6) Priority:2 NAL unit type:coded_slice packet payload length: 407	6) Priority:1 NAL unit type:coded_slice packet payload length: 407
7) Priority:2 NAL unit type:coded_slice packet payload length: 264	7) Priority:0 NAL unit type:coded_slice packet payload length: 303	7) Priority:0 NAL unit type:coded_slice packet payload length: 303
8) Priority:2 NAL unit type:coded_slice packet payload length: 286	8) Priority:2 NAL unit type:coded_slice packet payload length: 402	8) Priority:1 NAL unit type:coded_slice packet payload length: 402
. . .	. . .	. . .
A. Priority:2 NAL unit type:coded_slice packet payload length: 350	A. Priority:2 NAL unit type:coded_slice packet payload length: 350	A. Priority:1 NAL unit type:coded_slice packet payload length: 350
B. Priority:2 NAL unit type:coded_slice packet payload length: 235	B. Priority:0 NAL unit type:coded_slice packet payload length: 235	B. Priority:0 NAL unit type:coded_slice packet payload length: 235
C. Priority:2 NAL unit type:coded_slice packet payload length: 2035	C. Priority:2 NAL unit type:coded_slice packet payload length: 2035	C. Priority:2 NAL unit type:coded_slice packet payload length: 2035
D. Priority:2 NAL unit type:coded_slice packet payload length: 220	D. Priority:0 NAL unit type:coded_slice packet payload length: 220	D. Priority:0 NAL unit type:coded_slice packet payload length: 220

Table 4: NAL sequences comparison.

Thus, now all the NAL types are differenced by their priority, and the priority can be used to deliver the different frames to allow for an efficient differentiated service.

5. CHAPTER: HSDPA SCHEDULER STUDY

Once the video file has been properly encoded, the next step to develop is the implementation of the whole video transmission chain in the HSDPA network, in concrete over the system level simulator. In the first sections of this chapter, the way to set up video transmission chain will be explained, then the types of schedulers which have been implemented and finally the HARQ mechanism.

5.1. Video Transmission Chain

5.1.1. Elements

At the beginning of the work, the next three separated elements were available for implementing and improving the video transmission chain.

- Video Encoder
- Video Decoder
- HSDPA simulator

The encoder is a H.264/AVC encoder which produces a 264 file from a raw video file. The result is a file containing the NALs that carry the coding information. These NALs have all the information needed to reconstruct the video sequence at the decoder side. As the data is structured in NALs, it is ready suitable for transmission in packet switched networks.

The other video part is the decoder which from the 264 file is capable to rebuild the original video file and to give some information about the decoding process, such time to decode, type of frame, Y-PSNR, etc. This last parameter indicates the objective video quality.

The third and the most fundamental element is the HSDPA system level simulator. This tool allows simulating a whole HSDPA network at the MAC layer. The simulator generates a chosen number of users in a network sector with some determinate features depending on the random position of the users in the cell. The steps of this simulation are:

- Schedule a specific user
- Update the user position, the channel and transmission parameters associated
- Evaluate the BLER of the user from the SINR of the user channel
- Generate the ACK/NACK result for the current transmission from the BLER value
- Feedback the CQI (channel quality index) to the Node B function
- Update the monitor parameters and return to the beginning

Thus, the result of the simulation is a set of traces which inform about the behavior of the HSDPA network.

So, we have three separated elements, which have different inputs and outputs, and must be interconnected and improved to create the whole video transmission chain.

5.1.2. The interfaces

With the video file fixed, the next step to implement was to create the interfaces between the video encoder and the HSDPA simulator, and between the HSDPA simulator and the video decoder. This implementation was done with one customized Matlab script.

The encoder-simulator interface script reads the 264 file, differentiates the NALs, interpretates the NALs headers and creates a Matlab structure with this information. This structure contains the priority, the type and the length of the NALs. If the video receiver user is scheduled, the simulator reads this variable dynamically, to fit the NALs into TB (Transport Blocks) according to the encapsulation method explained in the third chapter. The reason to read this variable dynamically is to fit the size NALs with the variable size of the TB, which depends on the channel conditions (ACM).

During this process, the size of the SDUs is increased due to the addition of the lower layers headers. The simulator uses as import a new variable with the TB trace of the video user. This variable carries the following information; TB index, TB size, TB ACK/NACK information and NALs carried by the TB.

The script that manages the simulator-decoder interface is in charge of creating a received 264-video file. To do that, the original 264-video file is recollected depending on the content of the simulator video user trace. For example, if the TB that carries a determinate NAL has an ACK, this NAL is copied into the received file. In the opposite case, the NAL is not copied in the received 264 file simulating the loss of this video packet.

The following diagram summarizes the simulation process.

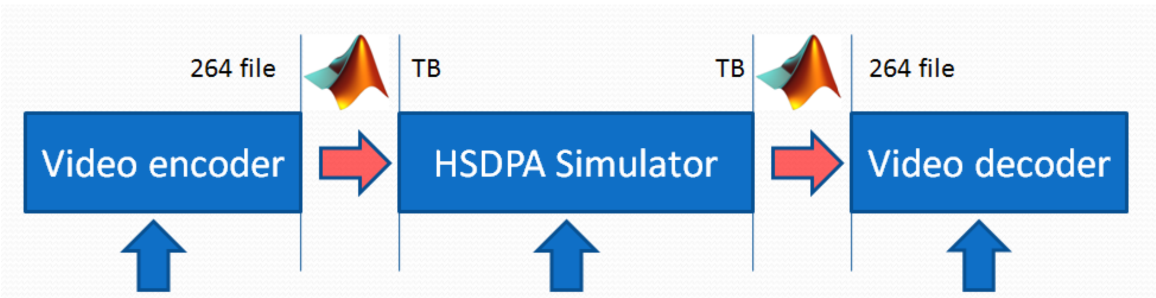


Illustration 22: Video transmission simulation chain.

5.2. Initial scheduler study

With the whole video transmission chain interfaced with the simulator, I concentrate my efforts on developing the video content aware scheduler. But first a study of the common schedulers was done to establish some basic understanding.

5.2.1. Conditions and parameters of the initial study

To realize this study, the position of the users in the sector was fixed in an uniform way, in order to take in to consideration the influence on the channel conditions and the increment of users in the cell. With increasing number of users, some of them could be in the worst cell positions basically far away from the antenna, affecting the entire simulation results. This placement effectively limits the length of the simulation trace needed to generate a sufficient averaging over the cell area. Furthermore, a specific user-positioning pattern leads to reproducible simulations results more easily. An example of this user's placement is shown in the next figure.

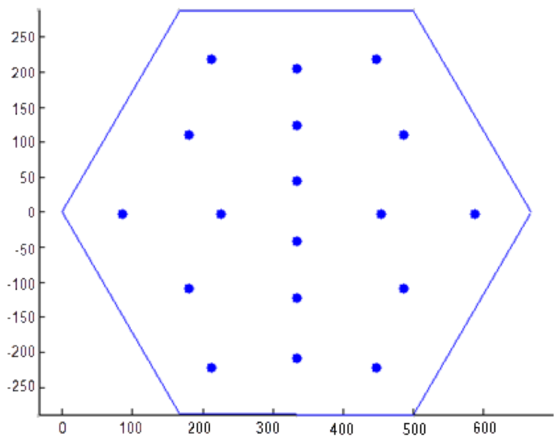


Illustration 23: Users uniform positioning for 18 users scenario.

The simulation time is other aspect to take into consideration in this study, which determinates the precision of the results. With a higher simulation time the network and the simulator reach the steady state which gives stable results. Also the number of users in the cell influences the results. With more users more time is needed to reach the stable state. Therefore, the simulation time depends on an established number of slots equal to 7500 times the number of users located in the cell.

Other HSDPA network settings used in these simulations tests are:

- All the users were simulated with the ITU Pedestrian A characteristics.
- The HSDPA network and the user terminals are equipped with MIMO capabilities.
- The same fading features were used for all simulations.

The measured parameters for comparing the behavior of the common schedulers with the content aware scheduler were:

- Average Node B BLER.
- Average Node B throughput
- Average Node B fairness.

The averages were done among the different simulations carried, with different amount of users simulated in the sector.

The BLER measures the percentage of damaged TB received by the users over the total amount sent by the Node B. This parameter depends on how the scheduler chooses the users to send them data depending on the channel conditions. For instance, if the scheduler under study always serves the user with the best channel, the BLER will be the lowest because the most of the TB are correctly received. In the wireless networks the target value for this parameter is 10% [12].

By monitoring the average throughput of the Node B, we note how the scheduler assigns the resources. If the scheduler selects the user which has the best channel, this user will receive more data than the others due to the AMC. Thus, the Node B reaches the maximum throughput capability. For the chosen settings, enabling MIMO capabilities, this value is 28 Mbps [12].

The BLER and the throughput parameters measure how the behavior of the schedulers in terms of amount data sent and maximum number of correct TB sent, but we furthermore need a parameter which measures how fair the users are treated in terms of data received by the scheduler. This parameter is called the data-rate fairness coefficient and is defined by the formula 1 [2]:

$$F_R(\Delta T) = \frac{\left(\sum_{m=1}^{M} R_m(\Delta T)\right)^2}{\left(M \cdot \sum_{m=1}^{M} R_m(\Delta T)^2\right)} \quad ; \qquad (1)$$

with M denoting the number of considered users, Rm(ΔT) denoting the data-rate the user m achieved in time interval ΔT. In this case, a fairness value of one corresponds to optimal fairness within a given time interval ΔT with respect to the defined criterion, indicating that all users received identical data-rates.

Also, I considered measuring the resource-fairness coefficient which weights how fair the scheduler allocates the users. But, I reject this coefficient because the future content aware scheduler serves the video user more times over the rest due to the priority schema implemented.

5.2.2. Schedulers under study

The scheduler has to select one or multiple users among the users that want to access to the channel. But for each user the channel varies continuously along the time domain and along the frequency domain. This results in the users having different channel conditions (ICI, fading, ...)

and achieve different data-rates for the same resources. Moreover, the different data types that the users want to receive or send complete the multi-user diversity concept.

This multi-user diversity can be seen as beneficial, because by exploiting the channel variation and the data diversity improves the performance rather than mitigate them. Thus, the scheduler's must select a user that can achieve a good data-rate rather than another one that wouldn't make the best of it at the present moment. On the other hand, allocating resources to users with good channel conditions means that users in disadvantageous conditions may be deprived of service.

The basic scheduler and the one that was implemented in the HSDPA simulator is the **Round Robin (RR).** This scheduler assigns cyclically the resources to the users regardless of the channel conditions, therefore an equal share of resources to each user is given. Other features which describe the RR are the low computational cost and the simplicity of the algorithm. But this scheduler has some problems due to the AMC of the HSDPA MAC layer, because the users do not experience the same end-user data-rate. Thus, the fairness data-rate coefficient measured in the Node B is lower than in other scheduling methods.

Another scheduler is called **Proportional Fair Scheduler (PF)** where one user at a time is selected for transmission on the available code resources [2]. With respect to the RR scheduling, this increases the system throughput while maintaining the long-term allocation fairness between users. The PF scheduler allocates the user m* who maximizes the ratio of achievable instantaneous data-rate over average received data-rate. This approach can be broadened to **Generalized Proportional Fair (GPF)** by introducing weighting factors α ∈ [0;∞) and β ∈ [0;∞) as follows [2]:

$$P_m^S(n) = \frac{(achievable\ \ instantaneous\ \ \ dataratefor\ \ \ user\ \ m\ at\ time\ \ n\ on\ PRB\ \ s)^{\alpha}}{(average\ \ datarate\ \ \ user\ \ m\ has\ received\ \ up\ to\ the\ time\ \ n)^{\beta}} \qquad (2)$$

$$m^* = \arg\max\{P_m^S(n)\} \qquad (3)$$

Where PRB denotes the Physical Resource Block, s the resource block index and α, β are weighting factors which balance the formula.

The numerator depends on the instantaneous channel conditions due to the AMC. With a higher CQI, good quality channel conditions, a modulation with a more symbols is going to be used, so the size of the TB sent in the time slot will be bigger. However, the denominator controls the fairness among the different users placed in the sector served by the scheduler. This term takes into account the data-rate reached by the users during the calculation time. In the HSDPA case, the data-rate of each user depends on the past channel conditions due to the AMC.

By modifying the weighing factors α and β various schedulers with different targets can be developed. By tuning parameters α and β in the equation, the trade-off between fairness and system throughput can be tweaked. Increasing α will increase the influence of the achievable instantaneous data-rate, which enhances the probability of a user currently in good condition to be scheduled. This result reaches a higher system throughput, but less allocation fairness and less data-rate fairness. Increasing β will increase the influence of the average data-rate, which increases the probability of a user with a low average data-rate to be scheduled. This results in higher data-rate fairness, but lower system throughput [2].

The effects described above can be identified by the following two settings for α and β:

- $\alpha \neq 0$ (e.g. $\alpha = 1$) and $\beta = 0$ corresponds to the **Max Rate (PF MR) scheduler**, where the user with the highest achievable instantaneous data-rate at time n is scheduled, since the denominator in the equation is equal for all users. The maximum system throughput is obtained at cost of the lowest fairness.

- $\alpha = 0$ and $\beta \neq 0$ (e.g. $\beta = 1$) schedules the user with the lowest average data-rate up to the decision time, i.e. equalizes the average data-rates of the users, since the numerator in the equation is equal for all the users. This results in a **maximum data-rate fairness (PF MF) scheduler**, but in a low system throughput.

Another option for the weighing factors is to choose α and β equal to one, defining the **Conventional Proportional Fair (PF CPF) scheduler.** This scheduler provides a good trade-off between allocation fairness and system throughput by utilizing the multiuser diversity.

5.2.3. Schedulers comparison

Looking at the plot that compares the BLER of the different schedules implemented at the simulator and explained in the last section, there are some significant results.

- If there are more users in the sector and closer to the sector edge, the BLER increases for all the schedulers under study, less for the PF MR scheduler since always selects the user with the best channel conditions.
- The PF MF scheduler BLER increases to higher values compared to the other schedulers. This tendency is due to the aim of this scheduler which tries to serve in the same amount of data all the users. Even when they are placed in points with bad channel conditions, but without selecting the users with the zero CQI feedback reports. The users situated close to the sector edge are selected more times as they receive small sized TB and with a higher probability of damage.
- RR and PF CPF have similar behavior in this BLER comparison.

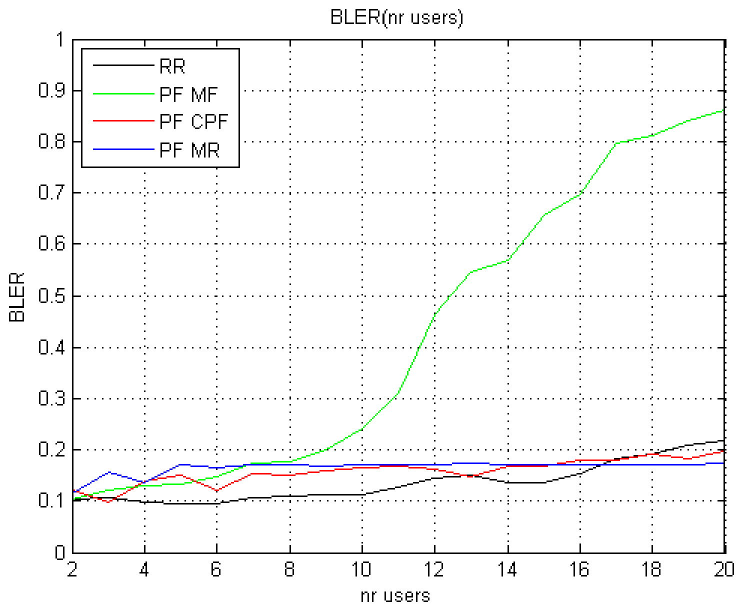


Illustration 24: Average transport block error rate looked in the node B depending on the number of users in the sector.

In the throughput plot, it is easy to differentiate the tendencies of the schedulers under comparison.

- The PF MR scheduler has a higher throughput performance, as the scheduler maximizes the Node B data-rate selecting the user that can receive the bigger TB as a result of the best channel conditions.
- Comparing the RR and the PF CPF scheduler, the performance of the last one is 1.8 Mbps higher than the RR due to the awareness of the channel quality that derives in a longer TB with more reliability.
- PF MF scheduler has the worst performance in terms of throughput. It selects the users with the worse channel conditions, but not the ones which report a zero CQI. In addition, these users receive the smaller TBs due to the AMC. The reason for this behavior is the aim of the scheduler, to be fair, delivering the same amount of data to all the users.

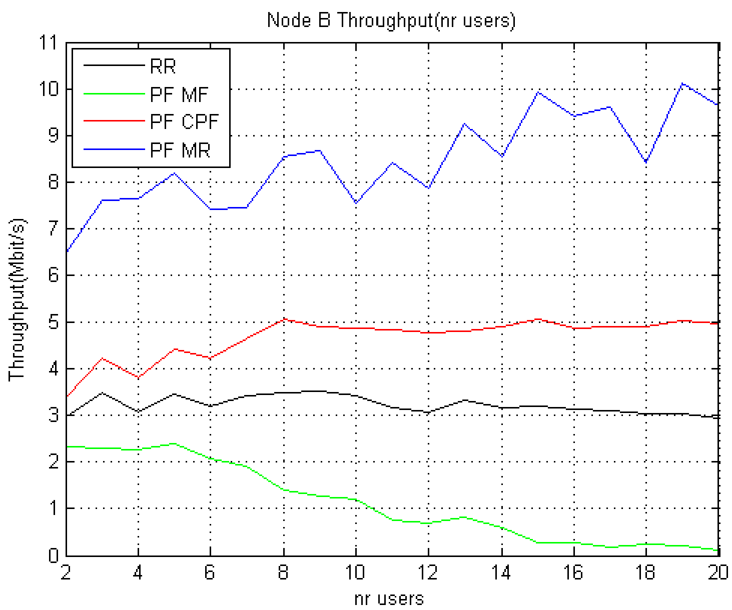


Illustration 25: Average node B throughput depending on the number of users in the sector.

The parameter, which completes this study, is the data-rate fairness coefficient. This value measures how the users are treated by the Node B.

- The value for the PF MF scheduler is the highest, 1. All the users have the same data-rate.
- The value of the fairness coefficient for PF MR scheduler decreases with the number of users in the network sector compared with the other schedulers. The reason of this decrement is the own aim of the scheduler, serving the users which can receive the longer TB, the users with the best channel conditions. Therefore, only the users that are in front of the antenna will be served.
- RR and PF CPF schedulers have similar behaviors, the performance is a little bit higher for the PF CPF. That means that the balance made by the PF CPF has the same behavior as the RR.

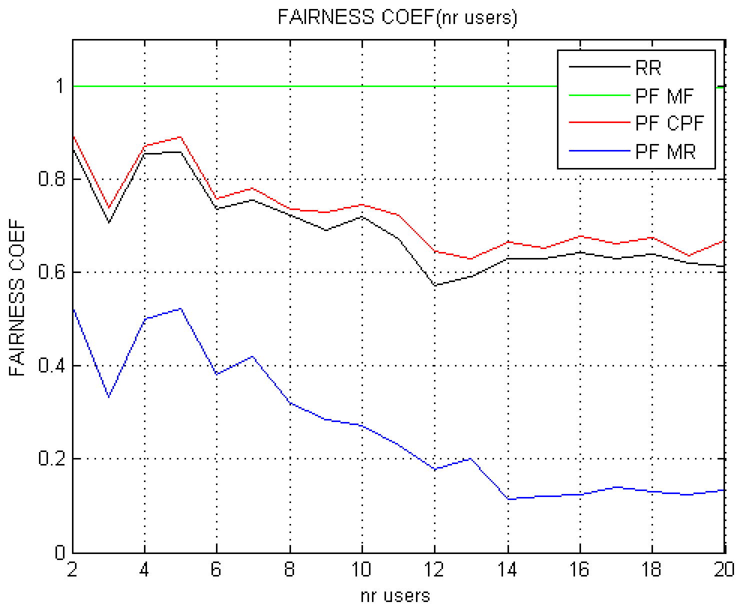


Illustration 26: Average node B fairness coefficient depending on the number of users in the sector.

The PF CPF is definitely the basis for the video content aware scheduler because:

- It has the balanced trade-off between fairness and performance.
- BLER and fairness coefficient are similar to the RR: the users are served in a similar way and the amount of damage TB is acceptable.
- The throughput performance is bigger than the RR (1.8 Mbps), by selecting the users that can receive larger TB due to the best channel conditions.

6. CHAPTER: HSDPA MODIFICATIONS

In this chapter, the proposed video content aware scheduler is explained. This scheduler is developed based on the CPF scheduler, but with the aim of keeping the recommended video data-rate. Also, a dynamic implementation of HARQ procedures is going to be described as well.

6.1. Content aware Scheduler

Given the results of the previous chapter, a novel video content aware HSDPA scheduler is designed. This scheduler completes the proposed cross-layer optimizations for mobile networks. It is based on the following arguments:

- The CPF scheduler is the ideal scheduler to be the reference for this novel scheduler. The reasons are the good results obtained in terms of fairness, throughput and NALER during the initial scheduler study.
- A recommended data-rate has to be maintained in order to feed continuously the decoder. Thus, the data-rate of the video streaming will avoid the diversity of requirements of the users placed in the sector served.
- The proposed scheduler has to maintain the same fairness values in the Node B, even when video prioritization is performed.

In order to explain the **mathematical model** that serves as a base for the video content aware scheduler is convenient to remember the CPF scheduler. This scheduler is based in the following formula (for more information see the previous chapter), where the weighting factors alfa and beta take the value of one.

$$P_m^S(n) = \frac{(achievable\ \ instantaneous\ \ \ datarate\ \ for\ user\ \ m\ at\ time\ \ n\ on\ PRB\ \ s)^{\alpha}}{(average\ \ datarate\ \ \ user\ \ m\ has\ received\ \ up\ to\ the\ time\ \ n)^{\beta}} \quad (4)$$

$$m^* = \arg\max\{P_m^S(n)\} \qquad (5)$$

Lets us note that the numerator depends on the channel conditions due to the AMC, where quality channel conditions set the size of the TB. Therefore, the alpha factor which weights the numerator value will be the factor which controls the priority information and the data-rate information. If the upper layer signalizes a high priority NAL, bigger alpha factor, it will be more probable to reach this user instead of others. Thinking on maintain the date-rate, if the video user has achieved a lower bit-rate than necessary to support the required bit-rate, it will be chosen to be served as soon as possible to solve the problem. Therefore, the alpha factor has the following definition, see formula 6, which depends on the data-rate recommendation and the priority signalization.

$$\alpha = \alpha^{(1)}(data\,rate) \cdot \alpha^{(2)}(priority) \qquad (6)$$

The denominator, on the other hand, controls the fairness among the different users placed in the sector. This term represents the average data-rate achieved for each user along the calculation time. Thus, setting a beta with the value of one is going to maintain the fairness condition in the decision formula.

6.1.1. Data rate term

The following picture shows the dependency of alpha on the data-rate, $\alpha^{(1)}$. This function has been optimized for the NALs which carry the P frames, being the most common in a video sequence. The blue line determines the recommended data-rate to feed the decoder properly, without overflowing the buffers but keeping them with enough data to decode the video sequence. The test case data-rate requirement is 220 kbps.

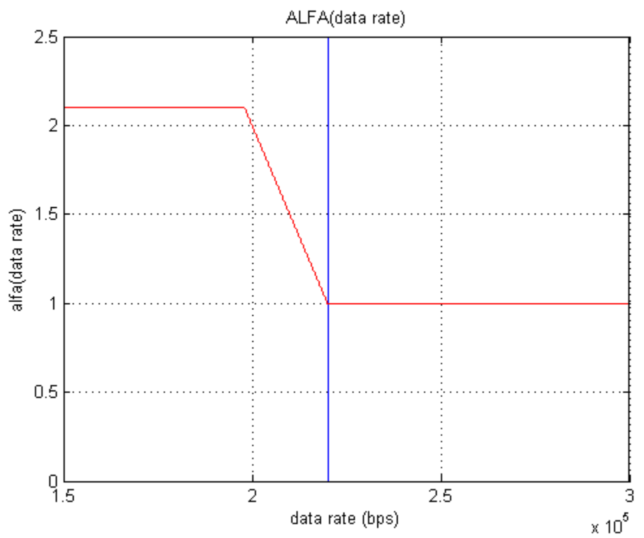


Illustration 27: Alfa depending on the data rate for the video content aware scheduler.

The red line assigns an alpha value to a specific data-rate. If the data-rate is bigger than the required, the behavior of the scheduler will be the same as in the CPF case. It is not necessary to distinguish the video user, if he maintains the data-rate. However, in the case of having a smaller data-rate than the required alpha is increased linearly with decreasing data-rate. The lower saturation point is set for a data-rate of 198 kbps (90% of the recommended). At this point, the alpha reaches a value of 2.1, which equals the following expression for the numerators of the scheduler's formula:

$$(\boldsymbol{min\ achievable}\ instantaneous\ datarate\ for\ \boldsymbol{a\ video\ user\ m}\ at\ time\ n\ on\ PRB\ s)^{2.1}$$
$$= (\boldsymbol{max\ achievable}\ instantaneous\ datarate\ for\ \boldsymbol{common\ user\ m}\ at\ time\ n\ on\ PRB\ s)^{1}$$

Therefore, if the scheduler has to choose between a video user, with a lower data-bit rate and a bad channel, and a user, who wants to receive background traffic with good channel conditions, the scheduler will select the video user, provided the same denominator for both of them.

6.1.2. Video priority term

In equation 6, $\alpha^{(2)}$ determines the dependency of the scheduling metric on the content priority, provided by the upper layer. The values of $\alpha^{(2)}$ as a function of the frame type are given in the table 5:

Frames type	Priority	Alfa(priority)
SPS	HIGHEST	1.4
PPS	HIGHEST	1.4
IDR	HIGHEST	1.4
I	HIGH	1.2
P	LOW	1
P_NON-REFERENCED	DISPOSABLE	0.8

Table 5: NAL types with their priority and alfa values.

A video user, which has to receive an important frame (I, SPS, PPS and IDR) will be scheduled almost immediately. That is like adding an offset to the alpha data-rate function designed for the P frame TBs. The resulting factor alpha is shown in illustration 28.

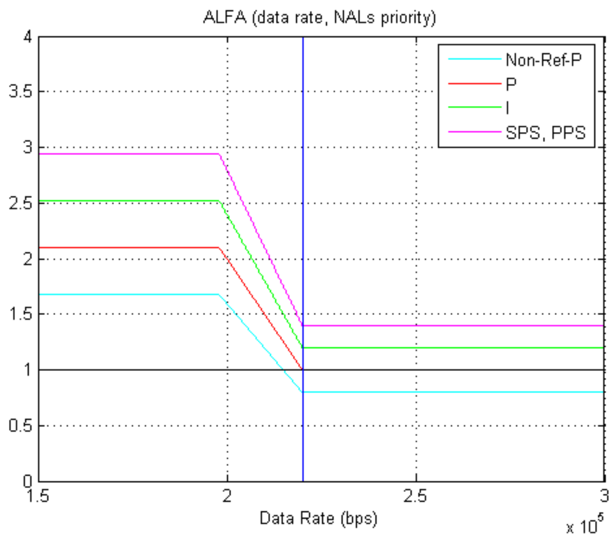


Illustration 28: Alfa depending on the data rate and the NALs priority for the video content aware scheduler.

7. CHAPTER: SIMULATION RESULTS

In this chapter I present the results obtained in the HSDPA simulator for the different cross layer implementations done during the research. The results are presented in two different blocks according with the investigation lines followed. The first block is referred to the comparison between the common schedulers and the video content aware draft scheduler. It is based only on the priority information provided by the layers higher than the MAC, where the scheduler acts. The second block compares the definitive video content aware scheduler with the reference scheduler, the CPF scheduler; the gain obtained by the HARQ method implementation is evaluated.

All of these simulation where referred to three particular user positioning scenarios, 6 users, 12 users and 18 users. The selection of these scenarios is due knows the behavior of the Node B when increasing of users on the cell.

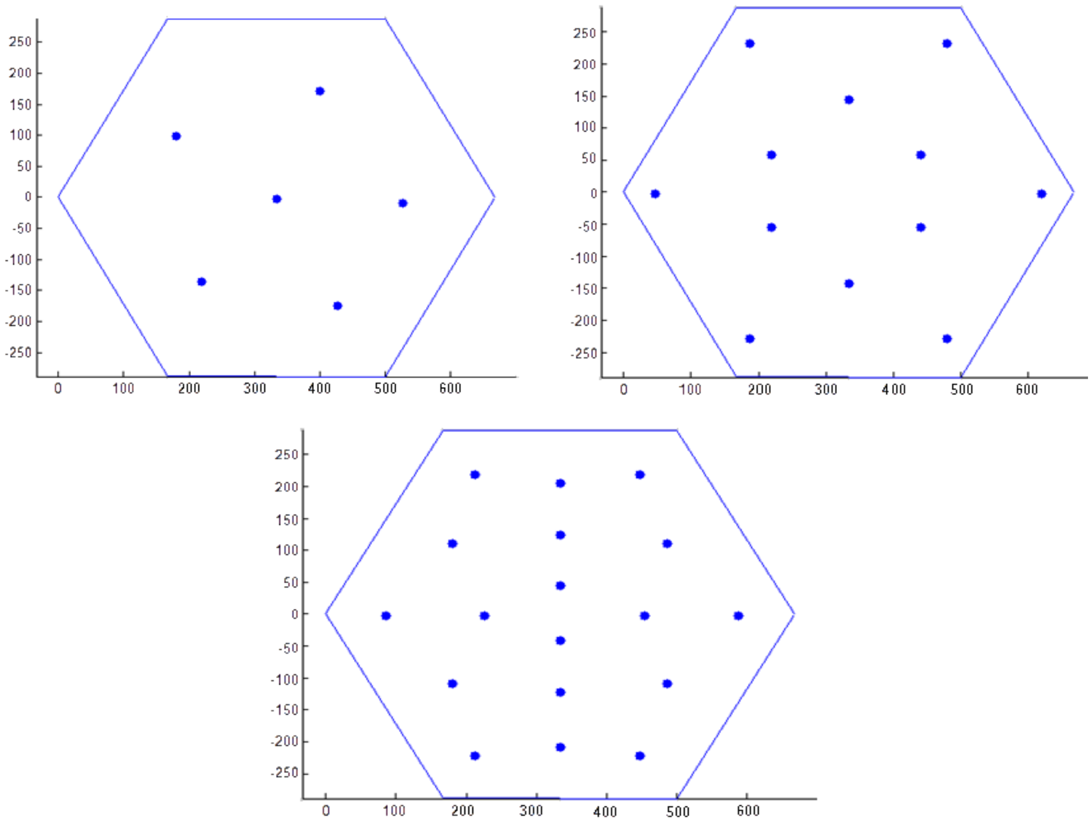


Illustration 29: Uniform users positioning for the 6, 12 and 18 scenarios.

Another restriction done when performing the simulations was to use only the SISO method. This decreases the complexity of the study and facilitates obtaining the results. The MIMO implementation for this specific case of study could be considered as future work.

7.1. Common scheduler vs Priority content awareness scheduler

The first designed scheduler was the one that is only aware of the priority information. The aim of this scheduler was to have a scheduler more advanced than the CPF, using the information obtained from the upper layers and with the encapsulation method implemented. This scheduler tried to reach the video receiver user as soon as possible in order to have enough time to request the possible retransmissions, helping the final HARQ method implementation.

Certainly, this is not the best policy to follow in a scheduler design because it prioritizes one single user over the others with a specific kind of data. But it was the first idea I had to mature the scheduler.

The results in terms of BLER were not as I expected because the TB error rate increased with the scheduler proposed. The video priority CA scheduler has 1.5% more BLER in all the scenarios than the CPF scheduler his reference. But this scheduler proposal maintains better performance compared with the others common schedulers, as you can see in the illustration 30.

The next result to discuss is the time required taken by the Node B to send the 30 seconds of video using the different schedulers. The average ones obtained by all the users in the cell at each location scenario in the case of the video priority CA scheduler is better, than the value obtained by the other scheduling methods. With this scheduler the video can be received three times faster than the round robin, seven times than the MF and between three and five times than the MR. But the gain achieved respect to the CPF is not too big, only 0.25 times better, as it shown in the picture 31.

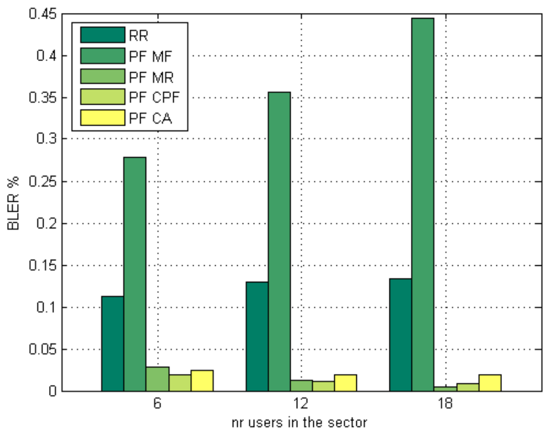


Illustration 30: Comparison of the average BLER at the Node B during the tx video simulation in the 6, 12, 18 scenarios for the schedulers under study.

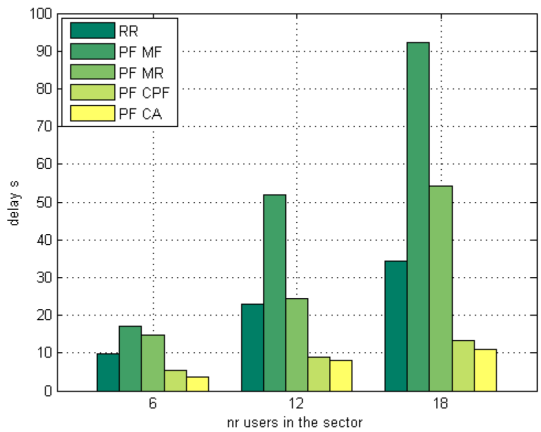


Illustration 31: Comparison of the average time needed to send the test video to a user in the 6, 12, 18 scenarios for the schedulers under study.

In the case the MF or MR schedulers are used, the users located in the cell edge need much more time than the average shown, and the users placed close to the Node B antenna need much less time to receive the video. Using MF, the users with the “bad” positioning are reached more often, but served with small TBs due to the policy to serve with the same data-rate and the AMC feature. In case of MR scheduler test, the difference in terms of delivery delay is bigger between “good” and “bad” positioned users, because the policy to serve the user with the best quality channel does that sometimes the edge users have been never served. With the other schedulers the deviation of the delay is not too high, because the users are treated with more equity.

The argument given in the last paragraph is not too objective, so in order to measure the equity among the users, I use the fairness coefficient to be more precise. In the illustration 32, you can see how the users, in the case of the MF scheduler are treated with the maximum data-rate equity; however the MR scheduler treats the users with less fairness. Video priority CA scheduler decreases the data-rate fairness respect to the CPF scheduler, which is its base. This decrease is due to the aim to serve the video receiver user with more priority than the others.

The throughput achieved by the Node B is shown in the illustration 33. The MR scheduler has the best performance and the MF the worse, such it can be predicted by the previous comments. If the user with the best channel (user that receives bigger TB) is selected in each TTI, the average data-rate of the Node B is the higher for the HSDPA network configuration set. In the case of the video priority CA scheduler the result is comparable with the CPF scheduler implementation, so this scheduler only determinates how the end-user is chosen and not how maximizes the Node B throughput performance.

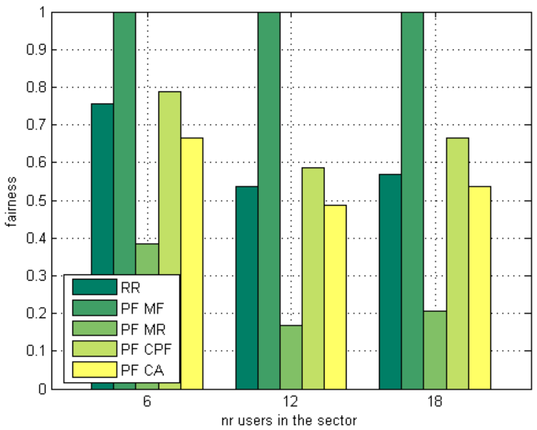


Illustration 32: Comparison of the fairness coefficient when the test video is sent to a user in the 6, 12, 18 scenarios for the schedulers under study.

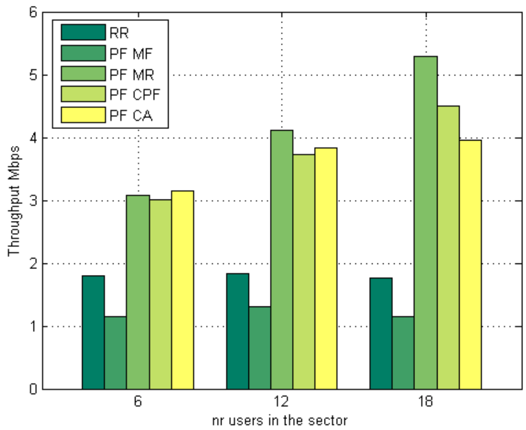


Illustration 33: Comparison of the average throughput reached by the node B when the test video is sent to a user in the 6, 12, 18 scenarios for the schedulers under study.

All the results explained in the previous paragraphs are referred to the network behavior, and the objective of this research is to increase the video quality at the end of the transmission chain. Therefore I compare in the following pictures, 34 and 35, the results obtained in terms of video packet error rate (NALER) and video quality (Y-PSNR).

The percentage of damage NALs for the round robin scheduler is around 13%, bigger than the BLER, due to the encapsulation of more than a single NAL in a single TB. In the MF case, the NALER is around 20%, because the scheduler tries to serve the users with the same fairness, even when they have bad channel conditions and in addition more possibilities to fail the delivery.

The size of the TB also influences this rate, according to the encapsulation equivalence between NALs and TB that is less than one to one. The opposite situation occurs in the MR scheduler case, the selected user is the one with the best channel condition, with the larger TB size to receive. Thus, a delivery failure has less probability to happen and more than one NAL fits into a TB (2 to 1 scale).

With the CPF scheduler only 2 out of 100 NALs arrive damaged at the end user. The reason to have this result is that more than one NAL fits into each TB, and the time slot selected to reach the user is the one with the best trade of between channel quality and fairness. These two factors decrease the probability of failure of a NAL transmission. However, the video priority CA scheduler has a worse performance, because it is 3 times higher than the CPF. Therefore, this implementation had to be matured and improved to decrease the NALER measured in the end user terminal.

The Y-PSNR comparison has a similar pattern to the NALER; the scheduler method with lower NALER is the one having the better quality and the other way round. There are some small differences in the values but they are caused by the type of frame lost, even inside of the average done. The lost of a TB with an I frame suppose a huge step backward in the quality values, as I have mention in the previous chapters. Note: the maximum Y-PSNR value for the test video is 35.6496dB.

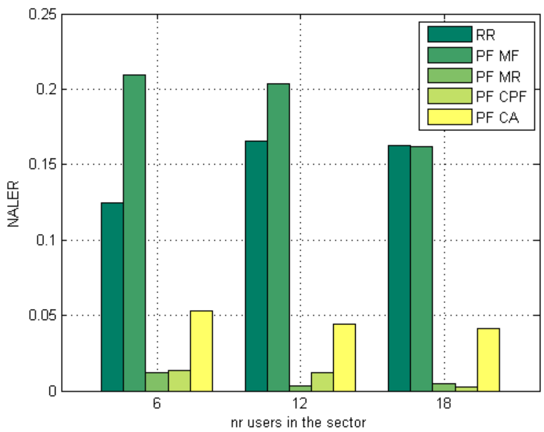


Illustration 34: Comparison of the average video packets error rate when the test video is sent to a user in the 6, 12, 18 scenarios for the schedulers under study.

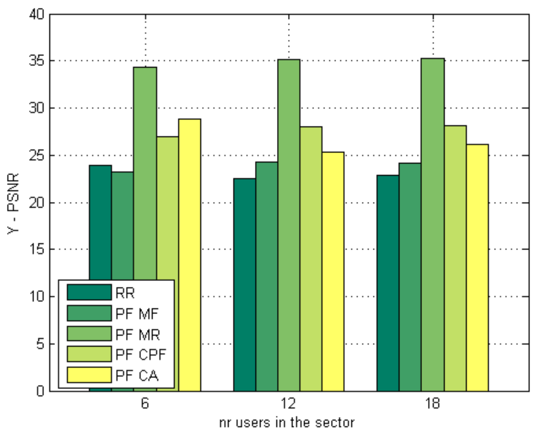


Illustration 35: Comparison of the average video quality, Y-PSNR, for the received video test to a user in the 6, 12, 18 scenarios for the schedulers under study.

In order to find a reason for the NALER increment in the video priority CA scheduler and mature the scheduler design, I plot the picture 38. This illustration shows in the upper graph the size of TB transmitted in bits versus the TTI index, and in the lower graph a one is plotted in case of damage transmission or a zero in the correct. Looking at the size graph I saw that the video priority CA scheduler selects the video receiver user in a burst sequence; and in the error plot the simulated channel produces the errors in the same. Thus, when these sequences match, the

NALER for the video receiver user increases more than the expected. The solution is the implementation of a flow control mechanism which avoids the burst transmissions.

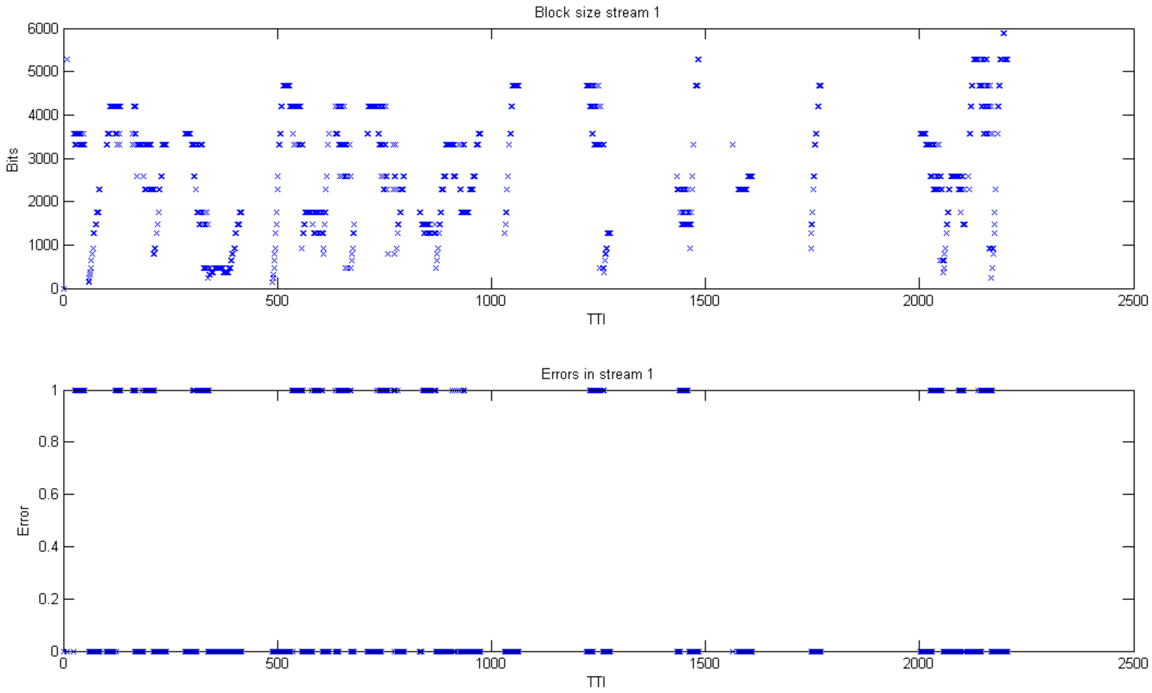


Illustration 36: TB size vs Time (TTI) and error vs Time (TTI).

7.2. CPF scheduler vs video content aware scheduler with HARQ

In this section, I compare the final results of the mature CA scheduler with the CPF. This scheduler has all the features explained in the chapter 6 and it completes together with the HARQ implementation the cross-layer design. But these improvements are not done at the same time, first I performed test with the HARQ implementation and then with the CA scheduler.

The average time necessary to send the test video increments with the HARQ implementation in the CPF case as I expected. Because more TTIs are needed to retransmit the damaged TBs. But in the case of the final CA scheduler with the flow control mechanism added, this time is almost constant, even with the HARQ mechanism and with the user increment in the cell. In our test video, this data-rate is 220000 bps and the recommended time is 8.4 seconds for the 231KB video file. This comparison can be seen in the picture 37, where all results include a 0.5 seconds delay to reach a stable status in the HSDPA network.

The flow control feature added to the CA scheduler makes the Node B fairer in the user treatment. This new scheduler do not prioritize video user over the rest, only gives him a special treatment in case the video user is served under the decoder recommended data-rate. Thus, the fairness coefficient is 3% higher than the CPF case, as it is shown in the picture 38.

The HARQ mechanism does not influence the fairness coefficient because the retransmissions are done in the case that user was selected, not immediately after the NACK report.

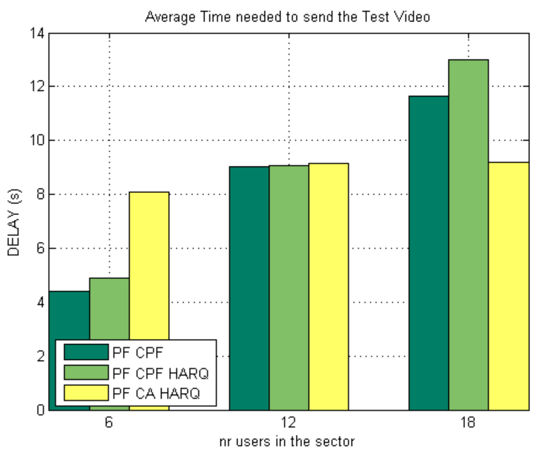


Illustration 37: Comparison of the average time needed to send the test video to a user in the 6, 12, 18 scenarios, with complete CA scheduler.

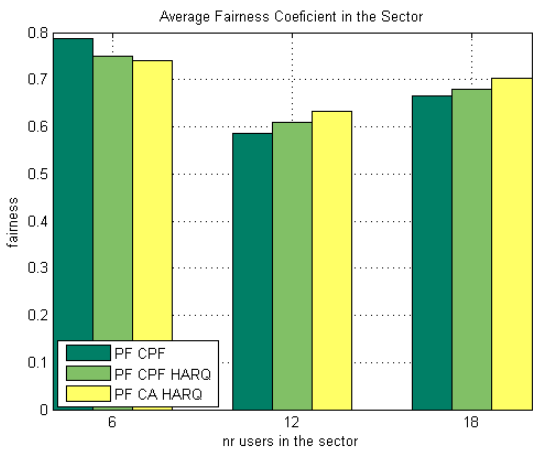


Illustration 38: Comparison of the fairness coefficient when the test video is sent to a user in the 6, 12, 18 scenarios, with complete CA scheduler.

With this new scheduler feature, the cross layer improves substantially the NALER. The HARQ mechanism decreases a 6% the video packet error rate and the scheduler another 2%, see picture 39. Only few NALs are received damaged by the end user and consequently the video quality increases.

At the same time, the Node B decreases its performance in terms of effective throughput (without retransmissions). The HARQ mechanism loses 0.5 Mbit/s over the effective throughput and the new CA scheduler another 0.5 Mbit/s in average (Illustration 40), due to the video receiver prioritization over rest.

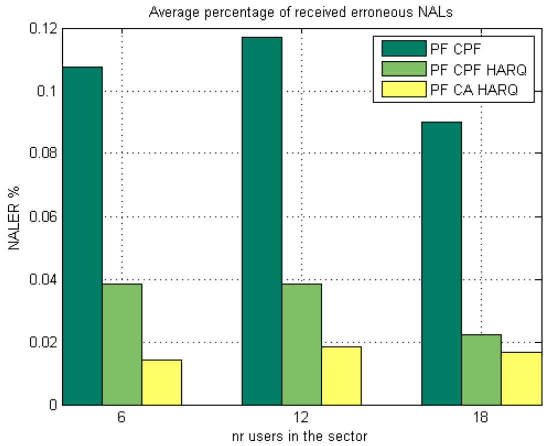


Illustration 39: Comparison of the average video packets error rate when the test video is sent to a user in the 6, 12, 18 scenarios, with complete CA scheduler.

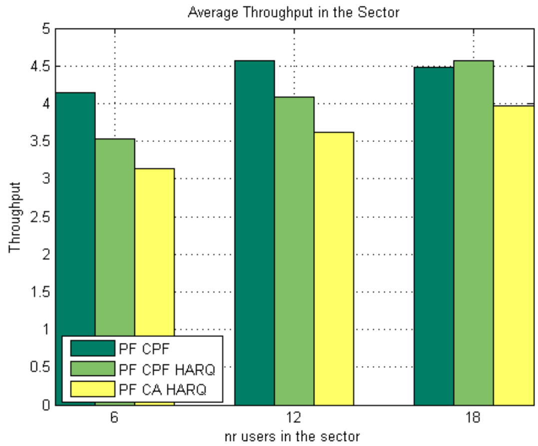


Illustration 40: Comparison of the average throughput reached by the node B when the test video is sent to a user in the 6, 12, 18 scenarios, with complete CA scheduler.

The special video receiver treatment given by the video CA scheduler achieves less damaged NALs, and therefore more objective video quality, as I show in the picture 41.

In concrete, the HARQ mechanism increases in average 5 dB the Y-PSNR and the CA video scheduler 1 dB more. Adding all the improvements, these almost reach the Y-PSNR measurement before the transmission, 35.56 dB.

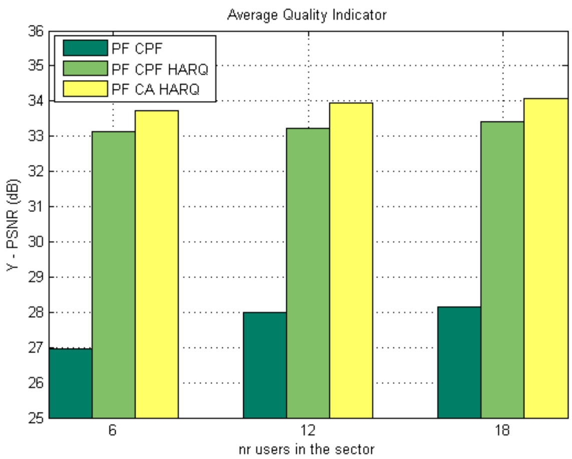


Illustration 41: Comparison of the average video quality, Y-PSNR, for the received video test to a user in the 6, 12, 18 scenarios, with complete CA scheduler.

To summarize the results, not only the quality has been improved. Also the CA video scheduler feeds with an almost constant bit rate the decoder, facilitating the data buffering. In terms of user treatment, the fairness coefficient does not decrease drastically with the new policy like the throughput.

8. CHAPTER: CONCLUSIONS & FUTURE ENHANCEMENTS

In this work I have made a study on how the video transmissions over the HDSPA mobile networks can be improved by a cross layer mechanism. The purpose was to study the error transmission patterns and implement an innovative video transmission chain in order to satisfy the user´s quality levels more efficiently.

To reach the objective, I modify two basic video transmission chain aspects, the video codification and the users scheduling. The basic H.264 encoder was modified in order to minimize the dependence among consecutive video frames, reducing the error propagation. At the scheduler I propose a content aware scheduler based on the conventional proportional fair scheduler. The scheduler has knowledge of which data types are sent to each user and which is the channel quality. These parameters allow the optimization of the user's selection to increase the video quality by working with the cross layer concept.

The path to achieve the proposed scheduler was based on the study the common schedulers by their implementation in a HSDPA simulator. Their modifications lead to the proposed CA scheduler which reaches a gain of 6 dB in the final video quality measured by the Y-PSNR and compared with the best common scheduler, CPF. However, this improvement is not totally due to the scheduler. An HARQ optimized to the proposed scheduler mainly gives the quality gain.

At the encoder side, we have introduced a new type of P frames, called non-reference P frames. These frames are inserted into the video sequence to reduce the decoding dependence. Their principle is: do not serve as a coding reference for the following P frames. Also a new frame identification scheme based on priorities helps to differentiate the frame types in a video sequence. At the encoder side, it allows different treatments to each frame type, giving the maximum priority to the I frames and the minimum to the new type, strengthening the content aware concept.

As next steps to increase the cross layer optimization, I suggest to investigate deeply a scheduler which mainly focused on the necessary data bit rate for each application to reach an acceptable QoS. This recommendation should also not lose the fairness perspective and the channel status knowledge.

BIBLIOGRAPHY

[1] Bader Al-Manthari, Najah Abu Ali, Nidal Nasser, Hossam Hassanein: “Generic Centralized Downlink Scheduler for Next Generation Wireless Cellular Networks”, ICC 2007.

[2] Christian Wengerter, Jan Ohlhorst, Alexander Golitschek edler von Elbwart: “Fairness and Throughput Analysis for Generalized Proportional Fair Frequency Scheduling in OFDMA”, PMCEL – MCOM, Panasonic R&D Center Germany GmbH, Langen, Germany.

[3] Sanjay Shakkottai, Theodore S. Rappaport and Peter C. Karlsson: “Cross-layer Design for Wireless Networks”, 2003.

[4] ITU-T: “H.264 Advanced Video Coding for Generic Audiovisual Services”, 2007.

[5] Shintaro Ueda, Hiroshi Shigeno, Ken-Ichi Okada, “Nal Level Stream Authentication for H.264/AVC”. IPSJ Digital Courier, Volumen 3, 2007.

[6] Thomas Wiegand, Gary Sulliivan, and Ajay Luthra. “Draft itu-t recommendation andfinal draft international standard of joint video specification (itu-t rec. h.264 — iso/iec 14496-10 avc)”. Joint Video Team, 2002.

[7] Iain E.G. Rchardson. “H.264 / mpeg-4 part 10 white paper - h.264 overview”, 2002. Available at: www.vcodex.com.

[8] Thomas Wiegand, Sullivan Gary, Gisle Bjøntegaard, and Ajay Luthra. “Overview of the h.264/avc video coding standard”. IEEE Tansactions on Circuits and Systems for Video Technology, 13:2, 2003.

[9] H. Holma and A. Toskala, “HSDPA/HSUPA for UMTS – High Speed Radio Access for Mobile Communications”, John Wiley & Sons, Ltd, 2006.

[10] M. Wrulich and S. Eder and I. Viering and M. Rupp, “Efficient Link to System Level Model for MIMO HSDPA,” Proc. IEEE 4th Broadband Wireless Access Workshop, 2008.

[11] M. Wrulich and M. Rupp, “Computationally Efficient MIMO HSDPA System-Level Evaluation,” EURASIP Journal on Wireless Communications and Networking, 2009, submitted.

[12] 3GPP, “Physical layer procedures (FDD)” (Tech. Spec. 25.214 V7.4.0), 2008. Available at: http://www.3gpp.org/ftp/Specs/htmlinfo/25214.htm

[13] 3GPP, “Medium Access Control (MAC) protocol specification” (Tech. Spec. 25.321 V8.5.0), 2009. Available at: http://www.3gpp.org/ftp/Specs/htmlinfo/25321.htm

[14] L. Superiori, M. Wrulich and P. Svoboda and M. Rupp, “Content-Aware Scheduling for Video Streaming over HSDPA Networks”, proposed paper to IWCLD, 2009.

[15] Xiaojun Lin, Ness B. Shroff and R. Srikant, “A Tutorial on Cross-Layer Optimization in Wireless Networks”, IEEE, 2007.

[16] Xue-Wu and Zhang Gang Hu, “Strategies of improving QOS for Video Transmission over 3G Wireless Network”, IEEE, Hohai University (Changzhou), Jiangsu province, China, IEEE, 2005.

[17] Network Working Group, “RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers” The Internet Society, 1998.

[18] Network Working Group, “RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP” The Internet Society, 2001.

[19] Network Working Group, “RFC 3190: RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio” The Internet Society, January 2002.

[20] F. Frederiksen and T.E. Kolding, "Performance and modeling of WCDMA/HSDPA transmission/H-ARQ schemes," Vehicular Technology Conference, 2002. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1040388

APENDIX A. ABBREVIATIONS

For the purposes of the present document, the following abbreviations apply:

ACK/NACK	ACKnowledge / No ACKnowledge
AL	Application Layer
AMC	Adaptable Modulation Coding
ARQ	Automatic Repeat Request
BLER	Block Error Rate
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
DSCH	Downlink Shared Channel
Diff Serv	Differentiated Services
GGSN	Gateway GPRS Support Node
GPF	Generalized Proportional Fair Scheduler
HARQ	Hybrid ARQ
HDTV	High Definition TV
HSDPA	High Speed Downlink Packet Access
HS-DSCH	High Speed Downlink Shared Channel
HS-PDSCH	High Speed Physical Downlink Shared Channel
HS-SCCH	Shared Control Channel for HS-DSCH
HS-SICH	Shared Information Channel for HS-DSCH
IDR	Instantaneous Decoding Refresh Picture
ISDN	Integrated Services Digital Network
JVT	Joint Video Team
L1	Layer 1 (physical layer)
L2	Layer 2 (data link layer)
MAC	Media Access Control
MIMO	Multiple Inputs Multiple Outputs
MPEG	Moving Pictures Expert Group
NAL	Network Abstraction Layer
NRI	NAL Reference Index
NUT	NAL Unit Type
PDU	Protocol Data Unit
PF	Proportional Fair Scheduler
PF CPF	Proportional Fair Conventional Proportional Fair Scheduler
PF MF	Proportional Fair Maximum Fairness Scheduler
PF MR	Proportional Fair Max Rate Scheduler
PHY	PHYsical layer
PPS	Picture Parameter Set

PT	Payload Type
QAM	Quadrature Amplitude Modulation
QCIF	Quarter Common Interchange Format
QoS	Quality of Service
RLC	Radio Link Control
RNC	Radio Network Controller
RR	Round Robin Scheduler
RTP	Real Time Protocol
SEI	Supplemental Enhancement Information
SDU	Service Data Unit
SNRY	Luma PSNR
SPS	Sequence Parameter Set
TB	Transport Block
TFT	Traffic Flow Templates
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
VCEG	Video Coding Experts Group
Y-PSNR	Luma Peak Signal to Noise Ratio

APENDIX B. HARQ IMPLEMENTATION

The Hybrid ARQ mechanism is a new feature established in the HSDPA standard. This mechanism is a variation of the ARQ error control method.

Standard ARQ consists on an addition of error-detection information (ED) bits to the data to be transmitted (such as cyclic redundancy check, CRC), in order to check the data at the receiving point and in case of retransmission. A transmission must be received error free on any given transmission for the error detection to pass. But in HARQ, could be also added some forward error correction (FEC) bits (such as Reed-Solomon code or Turbo code) to correct, if it is possible, the damage sequence. As a result of the increment of error detection-correction information the HARQ performs better than ordinary ARQ in poor signal conditions, but in its simplest form this comes at the expense of significantly lower throughput in good signal conditions.

At this point two different kinds of HARQ could be distinguish, type I adds both ED and FEC information to each message prior to transmission. When the coded data block is received, the receiver first decodes the error-correction code. If the channel quality is good enough, all transmission errors should be correctable, and the receiver can obtain the correct data block. If the channel quality is bad, and not all transmission errors can be corrected, the receiver will detect this situation using the error-detection code, then the received coded data block is discarded and a retransmission is requested by the receiver, similar to ordinary ARQ.

However, in Type II HARQ, the first transmission contains only data and error detection (no different than standard ARQ). If the PDU is received error free, it's done. If data is received in error, the second transmission will contain FEC parities and error detection. If received error free, it's done. If received in error, error correction can be attempted by combining the information received from both transmissions.

In practice, incorrectly received coded data blocks are often stored at the receiver rather than discarded, and when the retransmitted block is received, the two blocks are combined. While it is possible that independently decoded, two given transmissions are not possible to decode error-free, it may happen that the combination of all the previously erroneously received transmissions gives us enough information to correctly decode. There are mainly two ways of recombining in HARQ as has been explained in the second chapter.

Let me go into detail on HARQ is HSDPA: the data block is first coded with a punctured 1/3 Turbo code, then during each (re)transmission the coded block is usually punctured (i.e. only a fraction of the coded bits are chosen) and sent. The punctuation pattern used during each (re)transmission is different, so different coded bits are sent at each time. Although the HSDPA standard supports both chase combining and incremental redundancy, it has been shown that incremental redundancy performs almost always better than chase combining, at the cost of increased complexity, though.

HARQ can be used in stop-and-wait mode or in selective repeat mode. Stop-and-wait is simpler, but waiting for the receiver's acknowledgment reduces efficiency. Thus multiple stop-and-wait HARQ processes are often done in parallel in practice: when one HARQ process is waiting for an acknowledgment, another process can use the channel to send some more data.

The implemented HARQ scheme the Type II HARQ with incremental redundancy based on multiple stop-and-wait processes.

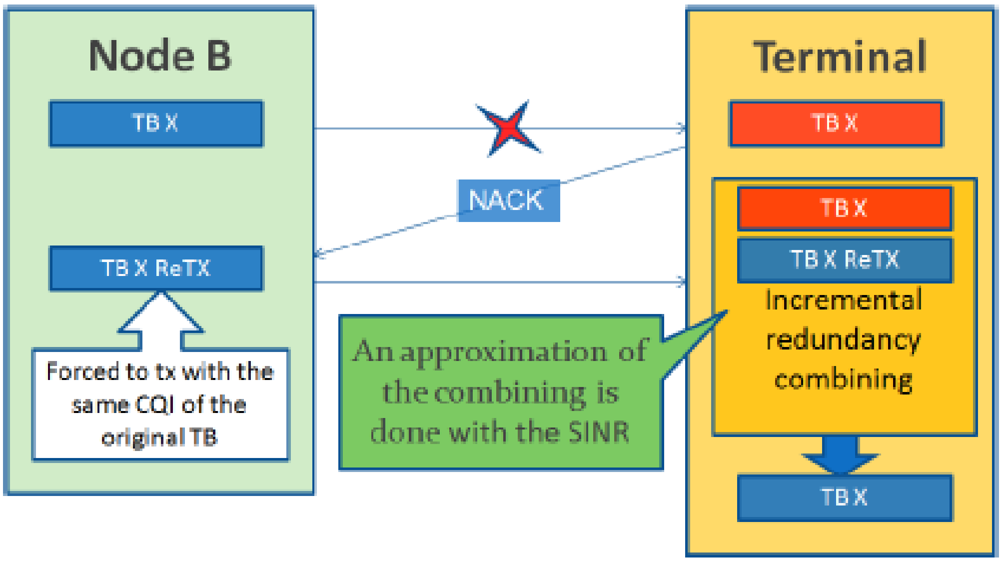
The Node B, the sending side of the transmission was programmed to create an HARQ process with each TB sent to a determinate user. These HARQ processes store the following information.

- TTI index: indicates the temporal timestamp of the TB.
- CQI: quality channel indicator of the transmission channel when the TB was sent. This value is important due to influence of it on the TB size (AMC).
- ACK/NACK value points which TB arrives correctly. If this value is 0 after the ACK/NACK t ransmission delay, the HARQ process for this TB is deleted. In the other case, the HARQ process will be activated to resend the erroneous TB.
- NAL identification index: this field contains the index of the video NALs carried by the TB, in order to resend the same NALs in the retransmissions. This field simplifies the NALs encapsulation and the desencapsutation in both sides of the communication.
- Size NAL: the NALs size of the NALs carried in the TB are indexed in this field. Also, it helps the encapsulation method.
- TB size: to check the size given to the retransmissions by the AMC compared with the original transmission.
- Current retransmission indicator: when the HARQ process involves a retransmission, this field indicates the number of retransmissions. The maximum value for this parameter is three.

Once an HARQ process is created, the node B sends the TB. On the other side of the communication, when the TB arrives, the user terminal checks the correctness and if a TB is OK an ACK is sent 4 TTIs after and the TB is passed to the upper layers. But, in case of a damage TB, this is stored in the terminal waiting for the retransmission. Also the user terminal sends a NACK message to the node requesting for a retransmission.

Four TTI after the TB delivery, the ACK / NACK message arrives at the Node B, and then the HARQ process is deleted in case of correctness or used to resend the NALs with the same CQI thereby increasing the current retransmission indicator.

At this point, when the retransmission arrives to the user terminal, this combines the original damaged stored TB and the retransmitted trying to obtain the correct data sent. But this combination is not done in fact in the simulation. The simulator compares the TB correctness by the comparison of the SNIR of the channel according to the explanation done in [20]. That means that the SINR value determinates the correctness of each TB and with each retransmission done the SINR for the retransmitted TB increases simulating the incremental redundancy. The last steps which explain the HARQ procedure are shown in the next picture.



Ilustration 42: HARQ function schema.