École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de Docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité: **Électronique et Communications**

JUAN CANTILLO

Codage multi-couches pour systèmes de communication par satellites

Soutenue le 19 May 2008 devant le jury composé de:

Gorry Fairhurst	Professeur, Université d'Aberdeen	Rapporteur
Michel Bousquet	Professeur, ISAE Toulouse	Rapporteur
Isabelle Buret	Responsable R&D, Thales Alenia Space	Examinateur
Jean Claude Belfiore	Professeur, ENST Paris	Examinateur
Marie-Laure Boucheret	Professeur, ENSEEIHT Toulouse	Directeur
Jérôme Lacan	Maître de Conférences, ISAE Toulouse	Co-directeur
Catherine Morlet	Ingénieur Système Télécoms, ESA	Invité

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Cross-Layer Optimization Techniques for Satellite Communications Networks

by

JUAN CANTILLO

A thesis submitted to the
École Nationale Supérieure des Télécommunications

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Digital Communications Group

Defense date: May 19^{th} 2008

Committee in charge:

Gorry Fairhurst	Reader, University of Aberdeen	Reviewer
Michel Bousquet	Professor, ISAE Toulouse	Reviewer
Isabelle Buret	R&D Manager, Thales Alenia Space	Examiner
Jean Claude Belfiore	Professor, ENST Paris	Examiner
Marie-Laure Boucheret	Professor, ENSEEIHT Toulouse	Advisor
Jérôme Lacan	Associate Professor, ISAE Toulouse	Co-Advisor
Catherine Morlet	Communication System Engineer, ESA	Invited

Acknowledgements

This document presents the research work done from November 2004 to February 2008 during my PhD thesis in Satellite Telecommunications at ENST-Paris. Carried out almost entirely in Toulouse, this work was funded and supported by Thales Alenia Space's R&D division for advanced satellite systems, under the supervision of Isabelle Buret. To her and to my advisors, Professors Marie-Laure Boucheret and Jérôme Lacan, I am extremely grateful for having given me the chance to start this thesis, and for having provided me with their scientific guidance and vast professional expertise during these three years and a half. Among these three excellent persons, I wish to express my special gratitude to Jérôme, both from a scientific and a personal standpoint. His receptiveness and sense of humor made me feel sincerely lucky to be his pupil and friend, and his acute technical insight made our scientific discussions at the warmth of several coffee cups extremely stimulating. Both my privileged interlocutor when problems of any kind emerged and the craftsman of many of the scientific aspects of this work, I owe him a lot.

I had the chance to be surrounded not by one, but by three different research teams during this endeavour. The professionalism and support from Thales Alenia Space's R&D engineers led by Isabelle was simply great. This work would not have been completed without their direct and precious help. In particular, Stéphane Combes allowed our work to be taken to the IETF, and simulation tools provided by Alain Ducasse shaped crucial aspects of our research effort. I learnt a lot from Fabrice Arnal, Katia Lecomte and Cédric Baudoin, whose encouragements, time and patience made them essential persons in Thales Alenia Space for me. Next, the people of ENSICA's Computer Science and Mathematics Department (DMI, current DMIA-ISAE) made me feel at home with them for almost two years and a half, from November 2004 to March 2007. Jérôme, Tanguy, Yves, Fabrice, Laurent, Pierre, Bernard & René, Tarek, Hervé, Ahlem, Manu, Amine... they all contributed with their humor, their scientific competence and their human qualities in making this time at ENSICA extremely productive and fun. Among these persons, I would like to express my special gratitude and admiration to Director Patrick Sénac for his unconditional support, enthusiasm and enlightened leadership. The confidence and energy with which he has backed my work and initiatives, added up with the invaluable pieces of advice I owe him have been extremely important for me during these years. Finally, the Telecommunications for Space and Aeronautics Laboratory Laboratory (TéSA) welcomed me from March 2007 until the completion of this work. As many of my colleagues there, I could hardly think of a more favorable environment to work than this refurbished old family house at the very heart of Toulouse, where more than a dozen nationalities coexist and research fields cover almost every area of digital communications. Many thanks to Marie-Josée and Sarah for their efficiency and diligent availability, and to Director Francis Castanié for having succeeded in forming and motivating such a nice, rich and competent team.

My gratitude also goes to Professor Michel Bousquet and to Dr. Gorry Fairhurst, for taking the time and patience to review this work. I could not have asked for a better jury. Undoubtedly, to

Michel and M. Maral I owe a deep love for satellite communications, injected through their superb and vivid classes at SUPAERO. I still use their slides and exercises for the classes I do myself. Gorry's support within IETF's IPDVB WG was paramount to take our work on DVB-S2 forward, and — perhaps without noticing it — he has taught me a lot about satellite engineering and team leadership during these years. I particularly value his ability to formulate accurate critics, to make the good questions and to value other people's points of view, all the while being modest despite his vast experience and knowledge of satellite networking.

Finally, what would this all be without the support and love from my family and friends? To Mariana and her love I owe some of my happiest moments in Toulouse; to Polo, Diego and Miles Davis some of my longest nights; to Daniel, Yannick, Sébastien, Marion, Alexis, Gael, Adriana, Benoît and Tomoe some of my greatest memories in Toulouse, Paris and Tokyo. Last but not least, I wish to express my boundless gratitude to my father and Claudia above all. Not only have they always supported me and inculcated me the love for study and knowledge, but they have especially been absolute models of intense courage, integrity and perseverance during my whole life. To them I dedicate this work.

Juan CANTILLO
Bogota, February 4^{th} 2008

Abstract

Current satellite architectures for delivering interactive IP services and broadband connectivity are based on the layered principles of the OSI reference model. There is no denying that the traditional research approach focusing on layer-specific problems faced by satellite architectures within the well-defined bounds of the layered model has been rather fruitful. Wireless-friendly adaptations of major protocols exist today, and state-of-the-art coding and modulation techniques have taken physical layers close to their theoretical performance limits. However, a number of critical issues such as end-to-end fulfillment of service-level agreements, seamless mobility or scalable support for reliable multicast have not yet found optimal solutions by means of independent layer tuning, due to the unique characteristics of satellite links. The modular approach blurs the dynamics of layers interaction with the wireless medium, hindering the overall system performance with redundancy, inefficient resource handling and suboptimal performances.

Recent research has thus started to address these problems in a holistic way, by stressing the potential benefits of authorizing information exchanges across layers beyond the scope of the reference model. Multi-layers feedback and the resulting system adaptivity offer multiple possibilities for attuning the protocol stack as a whole, allowing for overall optimization and better integration of satellite links in the increasingly heterogeneous network environment. Cross-layer design has emerged as a promising research area in the satellite and wireless communications fields, characterized by a multi-disciplinary approach involving information theory, network protocol design, optimization techniques, stochastic modeling and advanced signal processing. Since recent cross-layer proposals have started tackling successfully some complex problems that layered architectures do not address properly, next-generation standards and protocols are starting to integrate cross-layer principles *de facto*.

This thesis addresses the error control problem for satellite links from the perspective of cross-layer design. At the crossroads of QoS-related constraints, devices complexity and efficient spectrum use, error control is indeed a key aspect of wireless communications — particularly crucial in the satellite context — where cross-layer enhancements can play an important role. After a thorough introduction to cross-layer design, the first part of this work focuses specifically on the error control strategy of early DVB satellites, where redundancies between the channel decoder and the adaptation layers are set to light in order to propose a joint bandwidth-efficient error control policy. The focus then moves to second-generation DVB satellites and the definition of the novel, IP-centric and cross-layer friendly GSE encapsulation protocol, where results from the aforementioned study were successfully applied. Finally, a whole new cross-layer framework called HERACLES is introduced, offering efficient and overhead-free error correction capabilities for almost any layer of a protocol stack and being patented at the moment of writing these words. The results of the overall work show the strengths of an integrated approach to error control, and open the way for innovative cross-layer mechanisms to be deployed in next-generation communications networks.

Contents

List of Figures

List of Tables

List of Acronyms

This list summarizes the main acronyms used in this thesis. In general, acronyms are written in full words the first time they appear in the text, although some of them are repeated throughout the document (usually, at the beginning of a new chapter) for the sake of clarity.

ACM	Adaptive Coding and Modulation
ACK	TCP acknowledgement
APSK	Amplitude and Phase Shift Keying
ARQ	Automatic Repeat ReQuest
AWGN	Additive White Gaussian Noise
b	A bit. For example, one byte consists of 8b
B	A Byte. For example, 80b = 10B
BCH	Bose Chaudhuri Hocquenghem
BER	Bit Error Rate
BSC	Binary Symmetric Channel
BSM	Broadband Satellite Multimedia
CCM	Constant Coding and Modulation
CLD	Cross-Layer Design
CRC	Cyclic Redundancy Check
CSACK	Cross-layer TCP Selective Acknowledgements
CSI	Channel State Information
DAMA	Dynamic Assignment Multiple Access
DBP	Delay-Bandwidth Product
DUDE	Discrete Universal Denoiser
DVB	Digital Video Broadcasting
DVB-H	DVB standard for handhelds
DVB-RCS	Return Channel via Satellite DVB standard
DVB-S	First generation DVB satellite standard
DVB-S2	Second generation DVB satellite standard
DVB-SH	Satellite-to-Handheld DVB standard
ECN	Explicit Congestion Notification
EIoSS	Efficient IP over Second Generation Satellites
ESA	European Space Agency
ETEN	Explicit Transport Error Notification
ETSI	European Telecommunications Standards Institute
ETSW	Error Tolerant Scanning Window
FEC	Forward Error Correction
FER	Frame Error Rate

FMT	Fade Mitigation Techniques
GIST	Generic Internet Signaling Transport
GS	Generic Stream
GSE	Generic Stream Encapsulation
HERACLES	Header Redundancy Assisted Cross-Layered Error Suppression
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPDVB WG	IETF's IP over DVB Working Group
ISI	Input Stream Identifier
ISO	International Organization for Standardization
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
LT	Label Type
MODCOD	Modulation and Coding type
MPE	Multi-Protocol Encapsulation
MPEG2	Motion Pictures Experts Group 2
MTU	Maximum Transmission Unit
NPA	Network Point of Attachment
NSIS	Next Steps in Signaling
OAM	Operations And Maintenance
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PEP	Performance Enhancing Proxy
PER	Packet Error Rate
PID	Packet Identifier for MPEG2 flows
PRMA-HS	Packet Reservation Multiple Access with Hindering States
PSK	Phase Shift Keying
QEF	Quasi Error Free
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RS	Reed Solomon
RSVP	Resource ReSerVation Protocol
RTP	Real-Time Transport Protocol
RTT	Round-Trip Time
SAR	Segmentation And Reassembly
SISO	Soft-In Soft-Out
SNDU	Sub Network Data Unit
SP	Static Pattern
TCP	Transmission Control Protocol
TEI	Transport Error Indicator
TS	Transport Streams
ULE	Unidirectional Lightweight Encapsulation
UMTS	Universal Mobile Telecommunications System
VCM	Variable Coding and Modulation
VCP	Variable-structure congestion Control Protocol
VoIP	Voice over IP
XCP	Explicit Control Protocol

Nomenclature

We have compiled here the most important notations employed in Chapters 3 and 5. Although we would have preferred to keep those notations coherent throughout the whole document, this has not always been possible. Notations classically used for finite fields algebra an decoding theory have been used in Chapter 3, while the terminology and nomenclature used in Chapter 5 is mostly taken from the signal processing field.

Finally, some notations appearing briefly have been omitted for the sake of clarity.

Chapter 3

E_b/N_0	energy per bit to spectral noise density ratio (dB)
CRC_r	r-bit Cyclic Redundancy Check
$GF(q)$	Galois Field with q elements
C	systematic linear block code in $GF(q)$
n	codeword length
k	source message length
$r = n - k$	total number of added parity symbols in a codeword
d_{min}	minimum distance for the $C(n,k)$ code
$\lfloor a \rfloor$	greatest integer less than or equal to a
t	correction capacity of the $C(n,k)$ code
m	integer parameter for the $C(n,k)$ code
ε	q-ary crossover probability of the binary symmetric channel
$\overline{x}$	sent codeword of length n
$\overline{y}$	received message of length n
$\overline{e}$	error vector affecting the sent codeword $\overline{x}$ ($\overline{x}+\overline{e}=\overline{y}$)
$d(\overline{x},\overline{y})$	Hamming distance between vectors $\overline{x}$ and $\overline{y}$
P_c	probability of correct codeword decoding
P_w	codeword error probability
P_u	probability of undetected codeword error
P_d	probability of detectable codeword error
$\eta = P_u/P_d$	ration of undetectable to detectable codeword errors

Chapter 5

$\mathcal{C}$	generic communications channel
F	total number of symbols in a SP

L	total number of symbols in the sent message X
$\hat{\gamma}$	estimated value of a scalar γ
$S = (s_i)_{i\in[0,F-1]}$	Static Pattern of length F symbols
$X = (x_i)_{i\in[0,L-1]}$	sent message of length L symbols
$Y = (y_i)_{i\in[0,L-1]}$	received message of length L symbols
$Y_{/[i,k]}$	subsequence of Y consisting of elements $(y_i, y_{i+1}, ..., y_k)$
$Z = (z_i)_{i\in[0,L-1]}$	similitude measures between subsequences of Y and S
Ψ	metrics for Z
P_{cd}	probability of correct SP detection
P_{fa}	probability of false alarm
PSR	probability of Static Pattern Recovery
η	detection threshold
η_{opt}	detection threshold maximizing PSR
ε	cross-over bit probability (hard detection)
$\Phi = (\varphi_i)_{i\in[0,L-1]}$	real gaussian noise vector for the AWGN
σ^2	mono-lateral spectral noise density of the AWGN channel
(a, b)	modulation and soft decoding parameters in $\mathbb{R}^2$
μ_i	mean of z_i
σ_i^2	variance of z_i

Chapter 1

General Introduction

1.1 Background and Motivation

1.1.1 The Context

Cross-Layer Design (CLD) has become the new hype in wireless communications research. During the last years, an amazing number of works have started exploring the possibilities offered in network optimization by collaboration between layers beyond the limits of classical layered architectures. Supporters of the new current argue indeed that layered systems fatally fail in the task of delivering ubiquitous, *resource-efficient* and seamless differentiated Quality of Service (QoS) levels for IP services over wireless links, which is precisely what network users ask for today.

In satellite environments, bandwidth and power resources are scarce. It is therefore without surprise that satellite communications engineers and scholars mindful of the aforementioned trends have been particularly active in the areas of CLD. As a matter of fact, their efforts have paid off: many current proposals deriving from the new approach have demonstrated a great potential for improving the overall system while optimizing resource usage to a certain extent. Not only have they allowed minimizing the impact of the unique characteristics of satellite links on end-to-end communication, but they have also provided some tools for QoS provisioning and shown the way for smarter resource allocation.

1.1.2 Purpose of this Work

This thesis is yet another study devoted to satellite CLD. Its originality from previous works lies in that it specifically focuses on *transmission reliability*, a key aspect of satellite communications. More specifically, it addresses error control issues in DVB-S [1] and DVB-S2 [2] satellites from a cross-layer perspective, in an attempt to achieve better resource (bandwidth and power) usage while maintaining — or improving — the existing service. At the crossroads of QoS-related constraints, congestion issues, terminal complexity, power consumption and efficient spectrum use, error control casts a long shadow over the whole protocol stack. For this reason, it is a good candidate for cross-

layer enhancements.

QoS, Resource allocation and congestion & rate control are major areas where issues with layering have been long since pointed out. Without surprise, these are the topics that have received the greatest deal of attention from the CLD research community in the past years. In contrast, proposals dealing specifically with error control from a CLD perspective are rather rare. The "multi-layer reliability" concept introduced in Fabrice Arnal's PhD work [3] and the development of Fade Mitigation Techniques (FMT) [4] are some attempts to apply cross-layer techniques to error control.

This work does not intend to be a guide to cross-layer designers dealing with transmission reliability. Instead, its purpose is to identify small problems related to layering that can be tackled successfully with realistic cross-layer techniques in the satellite context, and to propose *ad-hoc* solutions enhancing overall functionality. Rather than covering a wide array of layering inefficiencies, we have preferred to dig deep in those we have encountered. Finally, we have paid special attention to the practical feasibility of our cross-layer proposals. Undoubtedly, these reasons account partially for the fact that the main results of this work have led to practical realizations, either embodied in patents or in contributions to standards.

1.2 Error Control

In order to fully understand what the main weaknesses of the layered approach to system reliability are, we provide here a brief overview of some of its most important factual and historical aspects.

1.2.1 Historical and Economical Aspects

The problem of combating transmission errors — either due to noise or other sources — is at the very heart of the Information Theory, whose bases were laid by H. Nyquist [5], R.V. Hartley [6] and especially C. E. Shannon [7]. According to Shannon, source coding (data compression) and channel coding (redundancy addition for error control) can be performed separately and sequentially while maintaining optimality, a result often referred to as the "separation theorem". Shannon determined the important theoretical limit concerning the foreseeable quality of digital transmissions by means of a Forward Error Correction (FEC) code, which remained to be found. His theoretical result represented a major scientific challenge for thousands of researchers and engineers because of the important economic implications at stake. Improving the correcting capabilities of a system means, with the same quality of received information expressed in terms of a tolerated Bit Error Rate (BER), enabling a transmission system to work in the most severe conditions. It is then possible to reduce the size of the antennas or the required transmitted power, thus impacting the overall mass & power budget of the spacecraft. In space systems (not only satellites, but also probes and so on), savings can be calculated in tens of millions of euros, since the weight of the equipment and the power of the launcher can thus be considerably reduced. In mobile cellular telephone systems and commercial satellite networks, improving the error-correction capabilities of the system also allows the operator to raise the number of potential users, to deliver more services over the same bandwidth and to save the terminal's power supply.

1.2.2 Error Control in Satellite Links

Modern error control policies in satellite systems are based on the superposition of compartmentalized error-control mechanisms at different layers. FEC codes at the physical layer constitute their core component. Link layers deal with resilient errors after FEC decoding with state-of-the-art Automatic Repeat ReQuest (ARQ) mechanisms, which detect and retransmit erroneous frames at the image of TCP. Finally, checksums and Cyclic Redundancy Checks (CRC) protect against erroneous packet reassembly, undetected FEC errors or random hardware malfunctioning at the middle and upper layers. CRCs are very important components of the two existing DVB-S adaptation layers: the Multi-Protocol Encapsulation (MPE) [8] and the Unidirectional Lightweight Encapsulation (ULE) [9].

There is no denying that radio specialists and protocol designers have succeeded today in addressing their specific problems regarding error control the best possible way up to now, given that all these independent techniques have reached today a quite high degree of maturity. In particular, the discovery of Turbo codes and the recent comeback of Low Density Parity Check (LDPC) codes have seriously closed in on the ideal code, taking FEC performance extremely close to Shannon's bound. Since FEC codes constitute the major components of error control techniques and the aforementioned complementary mechanisms (ARQ and CRCs) are mature today, few advances are expected in the years to come in this area by means of independent layer optimization.

However, inefficiencies that have long since been identified in the *overall* handling of the problem subsist. This, added up to the maturity of current techniques make a cross-layer approach to the problem all the more relevant.

1.2.3 Some Error-Control Inefficiencies in Layered Architectures

Below is presented a non-exhaustive series of inefficiencies of layered architectures linked to error control.

The first one is related the "security margins" first generation satellites integrate in their link budgets to cope with sudden channel fading. Precious dBs that could be used to increase coding rates or save battery power are most of the time wasted, given that fading events represent only a small percent of the transmission time, and only affect localized geographical zones.

On the other hand, applications that prefer to have partially damaged payloads delivered rather than discarded are dampened by the indiscriminate elimination of erroneous packets at the lower or middle layers. On top of decreased application performances, those retransmissions required to meet the non-requested reliability target provoke increased delay.

Given that TCP was designed to interpret segment losses as congestions, its normal behaviour has traditionally consisted in reducing sharply its transmission window to alleviate the network's assumed overload, and to resume operation in slow-start mode. In the satellite context, where losses are rather due to failed integrity checks than to congestions, TCP's behaviour is clearly unadapted: not only does it plummet the instantaneous bit rate — often stifling the application, but it also incurs excessive delays due to the required Round Trips Times (RTT) — counted in

seconds — to resume its normal operation. Different TCP flavors have been proposed to modify TCP's window reactions to losses, but only until recently real mechanisms have been proposed to tackle the root of the problem. This very well-known problem and its proposed solutions are described later in this dissertation.

Last but not least, less publicized is the inherent paradox of Shannon's separation theorem. Compressing at the source coder and then adding redundancy at the channel coder is "optimal" in the information theoretical sense, with long source coded blocks and channel coding using a sequence of random block codes with length tending to infinity. In practical scenarios, however, the situation is often different. Popular compression algorithms such as Huffman's [10] or Lempel-Ziv-Welch's [11][12] are so sensitive to channel errors that a single bit error can blow up the whole scheme, putting into question the net gain achieved after the necessary retransmissions. The emerging current of joint source-channel coding [13][14] addresses these issues by analyzing the synergies achievable by blurring on purpose the clear borderline that has traditionally existed between source and channel coding. Research in this area is particularly active in the satellite community, with remarkable initiatives such as the "Shannon mappings" [15][16] for instance. Although promising from a conceptual standpoint, joint source-channel coding seems quite hard to implement in real systems as of today, due to the magnitude of layer modifications it requires.

1.3 Contributions of this Thesis

1.3.1 Scientific Contributions

The work done in this thesis can be divided into two different yet complementary parts.

Work on DVB Adaptation Layers and the Generic Stream Encapsulation Protocol

The first part of this work focused on DVB adaptation layers, and more specifically on the definition of an optimal encapsulation protocol for IP over DVB-S2. The lack of such component at the beginning of this thesis, added up to DVB-S2's cross-layer friendly features — Generic Streams (GS) and ACM mainly — made DVB-S2 a fertile playground for developing CLD proposals. As a result, a big deal of the energy spent during the early months of this work was devoted to reflections on this particular topic[1].

As a first step, requirements for a new encapsulation protocol with innovative error-control aspects were identified in the DVB-S2 context, and submitted to public discussion as an Internet Engineering Task Force (IETF) *Internet Draft*. Three years of long commitment, intense discussions and participation from different organizations — such as the Digital Video Broadcasting (DVB) consortium and the European Space Agency (ESA) — led to the joint definition and final standardization of the resulting Generic Stream Encapsulation (GSE) protocol.

[1] For instance, Appendix A describes an early — but unpublished — attempt to address the IP over DVB-S2 encapsulation problem by means of an ambitious CLD approach called EIoSS. It is presented here for informative purposes.

The contribution of this work to GSE was twofold:

- First, it kicked-off its definition and standardization at the IETF, who played an important role in the overall process.
- Second, it contributed to the design of the protocol itself, by influencing the way Cyclic Redundancy Checks (CRC) were integrated in it. Our analyses showed in particular that the classical approach of appending a CRC per carried packet was not optimal in the DVB-S2 context. As a result, only fragmented packets carry CRCs in GSE, which saves around 4 bytes per carried packet. This leads to non-negligible bandwidth savings for future DVB-S2 links, quantified in around 10% for small packets — whose growing proportion account today for more than 40% of the exchanged packets in the Internet.

HERACLES: Header Redundancy Assisted Cross-Layered Error Suppression

The second part of this work was devoted to the development of HERACLES. Standing for Header Redundancy Assisted Cross-Layered Error Suppression, HERACLES is an innovative, standard-independent framework that can be used in any packet-switched digital communications system. It consists on a series of cross-layer functions implemented at the receiver to bring *overhead-free* delineation (also known as packet synchronization or flow delimitation) and error correction capabilities to packetized — and possibly erroneous — information flows. Instead of relying on added control information (such as synchronization pilots or parity symbols) at the transmitter, HERACLES exploits the natural redundancy existing among headers of packets belonging to an information flow. The mechanism utilizes data's structural redundancy to assess packet positions in the bit stream, and performs header bit corrections if desired, based on carefully weighted success probabilities.

When HERACLES' output is directed to the input of an appropriate FEC decoder, important synergies can be triggered. HERACLES behaves as an inner error correction code, providing the FEC decoder with a somewhat cleaner version of the data flow it would have received without HERACLES. In those common cases where the FEC decoder is working just below the limits of its functional domain, the small correction brought by HERACLES has the potential to make the FEC subsystem toggle from a non-decoding to a full decoding state. Computer simulations show that in many cases, enhancements up to 1 dB can be observed under realistic system configurations using common TCP/IP stacks.

HERACLES was fully developed during this PhD work and it has been protected by two patent applications filled by Thales Alenia Space in Q4-2007 and Q1-2008. It is still at its early definition stages, so we expect active research in this topic in the next years.

1.3.2 Publications, Patents and Related Documents

So far, material from this thesis has led to 4 technical reports [17][18][19][20], 5 successive versions of an IETF Internet Draft [21], 2 conference papers [22][23], 2 journal papers [24][25], 2 patents

[26][27] and has contributed to one European standard [28]. Presentations related to these documents were done in 2 international conferences, one workshop and 4 IETF meetings in Europe, Canada and USA.

1.4 Organization of this Document

This document is divided in 6 chapters, of which this general introduction is the first one.

Chapter 2 deals with Cross-Layer Design in general, not only from the perspective of error control. Origins and particularities of this new research area are presented based on historic and factual elements. The first part of Chapter 2 includes of course a short — but necessary — reminder on classical layered architectures, and introduces the generic satellite model used throughout the whole document. Next, shortcomings of layered architectures are presented in the satellite context, completed by an overview of cross-layer proposals that have started addressing them. Finally, a discussion on CLD's perspectives, promises and risks concludes this introductory chapter.

Chapter 3 summarizes the first part of the work done on DVB adaptation layers, where no standard adaptation layer had been defined for DVB-S2. It was motivated by the general will to achieve an efficient adaptation/encapsulation protocol that would take advantage of the enhanced physical layer of DVB-S2, and especially of its stronger FEC scheme. The first part of Chapter 3 focuses on the way FEC and MPE/ULE's CRCs interact to achieve error control in the DVB-S context, and proposes simple cross-layer solutions to cope with the highlighted inefficiencies. The focus then moves to DVB-S2, where similar analyses and conclusions are drawn. Finally, GSE's design choices for error control are analyzed under the lights of this chapter's results.

Chapter 4 presents the second and final part of the work done on DVB adaptation layers. It specifically describes the motivation and rationale behind the definition of GSE, and the protocol itself. The inadequacies of MPE and ULE in the DVB-S2 context, GSE's design choices, its header formats and unique characteristics are fully detailed in this chapter.

Chapter 5 is devoted to HERACLES. Given that many mathematical notations and theoretical concepts are explained here for the first time, it is perhaps the most dense chapter of this document. After a detailed introduction to redundancy in packet-switched systems, the generic model for HERACLES is introduced. Next, applications under hard and soft detection configurations are considered, and results for both cases are derived. Finally, particular cases where important synergies can be achieved between HERACLES and appropriate FEC decoders are analyzed.

Finally, Chapter 6 closes this dissertation with a set of general conclusions, remarks and personal thoughts on the work done.

Chapter 2

Cross-Layer Design for Satellite Networks

2.1 The Basics

2.1.1 A New Approach to Network Design

Of late, a silent revolution has started shaking the foundations of three decades of network design. As wireless communications and technologies gain increasing importance, researchers have started to put the basic principles of layered architectures under close scrutiny. They argue that when addressing the challenges of today's networks, continuity solutions based on the Open Systems Interconnection (OSI) reference model are ill-fated. There is no denying that the layered model has served well for wired networks in "best-effort" mode, and that its principles have provided solid bases for sound software and network design. However, things have gotten considerably trickier with the increasing demand for end-to-end Quality of Service (QoS) solutions in heterogeneous and crowded network environments, with important wireless segments that — more often than not — constitute the bottleneck of the overall link.

Abandoning the sacrosanct rule of independent and modular protocol design, proposals violating on purpose the premises of layered architectures have started to appear, attempting to exploit dependencies between layers to reduce inefficiencies and thus achieve performance gains. Initiatives in this sense have originated from a wide array of organizations and researchers, who announce the need for a new approach to network design. Under the name of Cross-Layer Design (CLD) the new tendency has permeated in a few years almost every sphere of modern wireless communications, among which ad-hoc networks, wireless sensors and satellites.

In order to fully understand what is at stake with satellite CLD and how accurate the critics of the layered model are, the first part of this introductory chapter provides a reminder on the basics of layered architectures and satellite networks. Next, a detailed — but not limited to error control — presentation on the major shortcomings of layered architectures is presented. This gives the bases for a thorough review of CLD and its origins, completed by a comprehensive survey of the existing proposals for satellite and wireless networks. Finally, a discussion on CLD perspectives, promises

and risks concludes this introductory chapter.

2.1.2 Layered Architectures

The OSI Reference Model

The Open Systems Interconnection (OSI) reference model [29] is the embodiment of layered architectures. It is an abstract description for communications and protocol design for computer networks, developed by the International Organization for Standardization (ISO) in the late 1970s. Initially assorted with a set of concrete ISO protocols, it was intended to serve as the foundation for the establishment of a widely-adopted suite of protocols for international internetworking. However, the project never achieved widespread success: several factors among which the rise in popularity of the Internet favored the progressive adoption of the TCP/IP protocol suite instead [30]. Nonetheless, the architectural principles of the OSI reference model summarize the basics of almost all current communications networks (including TCP/IP-based ones), for which it is taught in every computer communications course and widely used for reference.

Application	Applications and application interfaces for OSI networks. Provides access to lower layer functions and services.
Presentation	Negotiates syntactic representations and performs data transformations (e.g. compression and code conversion).
Session	Coordinates connection and interaction between applications. Establishes dialogue, manages and synchronizes direction of data flows.
Transport	Ensures end-to-end data transfer and integrity across the network. Assembles packets for routing by the network layer.
Network	Routes and relays data units across a network of nodes. Manages flow control and call establishment procedures.
Data Link	Transfers data units from one network to another over transmission circuits. Ensures data integrity between nodes.
Physical	Delimits and encodes the bits onto the physical medium. Defines electrical, mechanical, and procedural formats.

Figure 2.1: The OSI Reference Model.

The OSI model abstracts features common to every communicatios system, and organizes them in a vertical set of 7 modules or *layers* shown in Figure 2.1, where each module provides services to the one above it and receives services from the one below *only*. Strict layering, the foremost important concept behind the OSI reference model, advocates tasks division into logical entities with standard interfaces, thus encouraging simplicity of operation and interoperability.

The TCP/IP Model

The original 4-layered TCP/IP model (a.k.a. the Internet reference model) was originally developed for the ARPANET — the predecessor of the Internet — by the US Department of Defense some years before ISO's initiative. The name comes from its two most important protocols, the Internet Protocol (IP) and the Transmission Control Protocol (TCP), true building blocks of the current Internet. The TCP/IP protocol suite is currently maintained by the Internet Engineering Task Force (IETF), who has deliberately avoided strict layering of all sorts in its official documents. Early TCP/IP's architectural guidelines such as RFC 1122 [31] and RFC 1958 [32] certainly refer to layers, but less formally and rigidly than ISO does, emphasizing practical and sound engineering principles over abstract layering[1].

For practical purposes, the 4 layers of the TCP/IP protocol suite can be mapped more or less accurately to the 7 layers of the OSI model as shown in Figure 2.2.

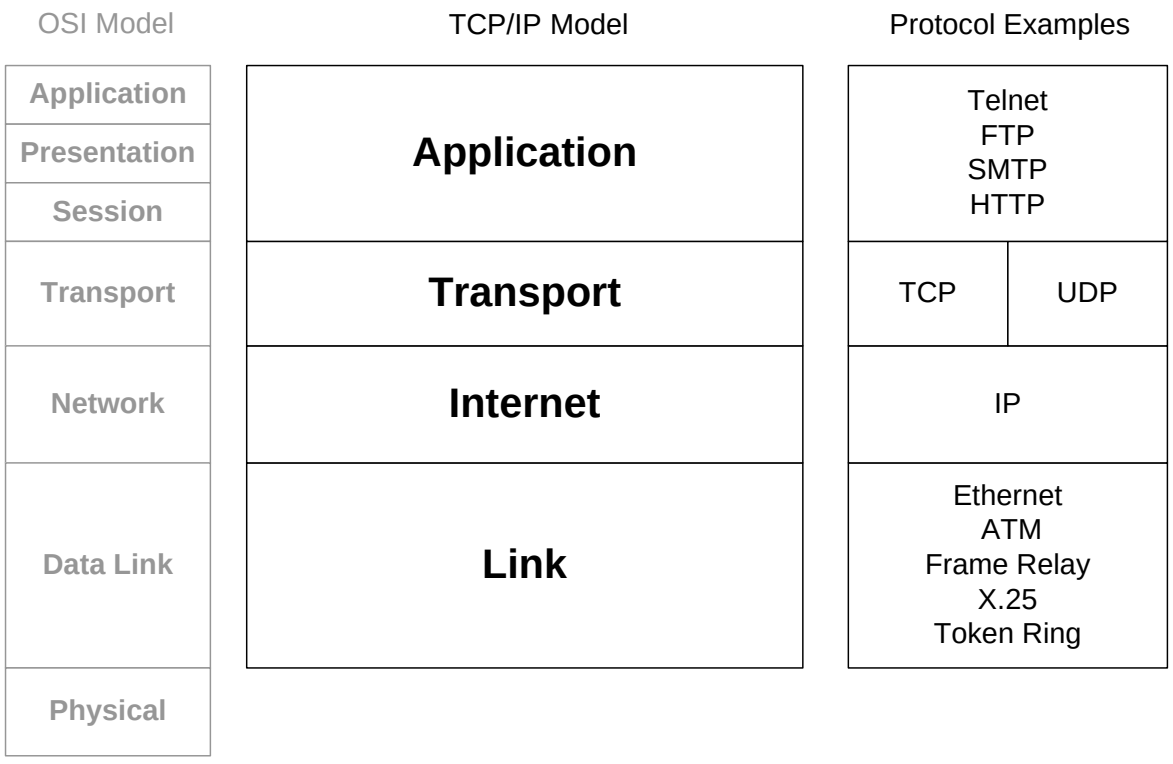


Figure 2.2: The TCP/IP Model, and an approximate correspondence between its layers and the OSI model (left).

[1] As a matter of fact, and probably because TCP/IP was already in use when the OSI model came out, the IETF has never felt obliged to be compliant with it. This position is stressed in the section "Layering Considered Harmful" of its more recent architecture document, RFC 3439 [33].

2.1.3 On Satellites and Satellite Networks

Satellite Specificities

In the following paragraphs, we focus on those particular aspects of satellite links affecting end-to-end communication. They are taken directly from RFC 2488 [34], which deals with satellites from a networking standpoint. Detailed description of satellite systems and links are out of the scope of this document, and can be found in [35].

From a networking perspective, the main technical issues with satellite links are:

Delay: Geostationary satellites, which are the main focus of this thesis, are located at an altitude of approximately 36 000 km over the equatorial plane [35]. The propagation time for a radio signal traveling twice this distance (corresponding e.g. to a single-hop link between two ground stations below the satellite) is around 240 ms, and rises up to 280 ms for ground stations located at the edge of the view area due to the increased satellite distance (41 750 km). Therefore, the propagation delay for a message and the corresponding reply (Round Trip Time or RTT) could be at least 560 ms, without taking into account other factors such as gateways queuing, processing delays etc.

Noise: Given that the amplitude of radio waves decreases in proportion to the square of the traveled distance, inbound and outbound satellite signals become very weak before they reach their destination. The situation is worsened by a multitude of factors such as shadowing or rain — particularly important for Ka Band (30/20 GHz) operations. From a practical standpoint, these factors bring about low signal-to-noise ratios that induce errors in the received bit sequence, characterized by the achieved Bit Error Rate (BER). Typical BER values for a VSAT front-end at the sub-satellite point range from 10^{-9} (nominal conditions) to 10^{-2} (functioning domain limits) for state-of-the-art hardware and technology. In order to meet tighter error control requirements at the upper layers and in order to provide service continuity guarantees, error control techniques such as Forward Error Correction (FEC) are used.

Resource constraints: The term "resources" in the satellite context refers to one (or a subset) of the following: its embarked power, the radio capacity of the satellite payload and the portion of the radio-frequency spectrum it can use, often referred to as *bandwidth*. The first two resources are limited mainly by technological constraints that dictate e.g. the maximum size and weight of the on-board batteries or solar panels, or the degree of miniaturization of its circuits. The radio spectrum, however, is a scarce and shared natural resource, controlled by international regulatory bodies (e.g. ITU) ensuring fair and long-term access for all. Bandwidth scarcity makes it difficult to trade it to solve other design problems, for which optimization efforts in this sense are important. Most of the work done in this thesis attempts to achieve resources — and specifically bandwidth utilization — gains.

From a networking point of view, the 3 aforementioned specificities of satellite links have important implications — regardless of the specific protocol stack used. The most important ones are listed below.

Long Feedback Loops: Due to the long RTTs involved in satellite links, mechanisms relying on

any kind of sender feedback — such as acknowledgements or retransmission triggers — are hindered, which especially affects interactive applications. On top of that, when delivering IP services over satellite, some of the TCP congestion control algorithms can be particularly affected by this (see Section 2.2.2).

Large Delay × Bandwidth Product (DBP): The DBP defines the amount of data a protocol should have "in flight" (data that has been transmitted, but not yet acknowledged) at any one time to fully utilize the available channel capacity. Practically speaking, this value defines the minimum size of the buffer a receiving protocol entity has to set — e.g. TCP's receiving window — in order to achieve maximum *throughput*. Note that when ensuring reliable data delivery, this value also determines *de facto* the amount of yet-unacknowledged data to be duplicated in a buffer memory in case the retransmission of a lost packet is required by a client. Satellite DBP magnitudes are in the orders of 10^7 or 10^8 bits, whereas for terrestrial links such as ADSL lines they are in the order of 10^4 bits[2].

Transmission Errors: Bit errors at the receiver front-end cause packet drops, leading to missing protocol data blocks. How badly this affects the overall communication depends on several factors, especially the tolerance to errors of the upper layers of the protocol stack. Although transmission errors remain a major issue for mobile terminals, the sophistication of current transmission techniques has greatly reduced the impact of this particular point for fixed satellites services.

Link Asymmetry: Typical forward links have greater capacities than return links (either by terrestrial or satellite channels, when available). A legacy from classical broadcast designs, this is partly due to the important difference in radio performances (antenna sizes, radioed power) existing between transmitting gateways and typical receiving terminals.

Note that satellite networks are not the only environments where the above characteristics are found. These are common problems of wireless systems in general.

Generic Satellite Reference Model Used in this Work

ETSI's Broadband Satellite Multimedia (BSM) Working Group has recently specified a reference architecture model for IP-based satellite networks [36] that could have been used in this thesis.

Given that this work focuses on error control aspects, we have preferred a simpler satellite model presented in Figure 2.3, that matches with more realism the actual layers implemented in satellite systems relying on DVB standards. From a practical standpoint, this is the model implicitly used by most researchers and designers of lower satellite layers and protocols.

In the rest of this work, satellite layers will sometimes be referred to with their associated number (e.g. L2 for the link layer) or their name, depending on the context. Figure 2.3 also introduces common names for the types of packets dealt with in every layer that we will use throughout this thesis. For example, Protocol Data Units (PDU) represent L3 packets of any kind, especially IPv4

[2] The delay used in this equation is the RTT and the bandwidth is the capacity of the bottleneck link in the network path.

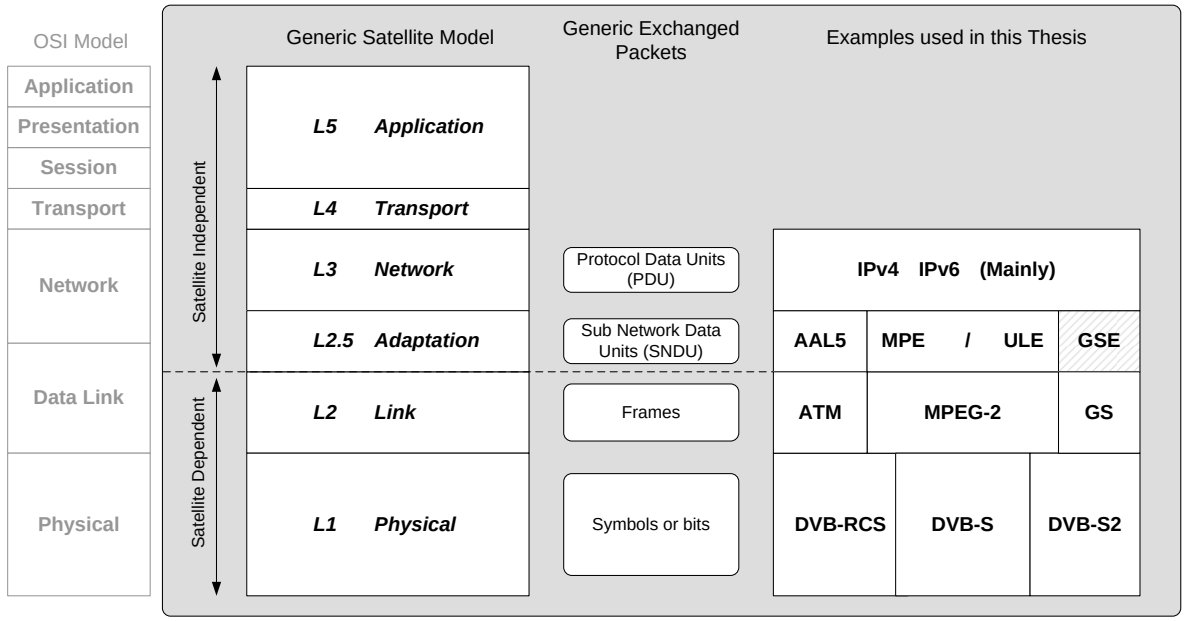


Figure 2.3: Generic satellite model used in this thesis.

and IPv6 packets. Sub Network Data Units (SNDU) correspond to encapsulated PDUs at the adaptation layer (L2.5) and so forth.

2.2 Understanding Cross-Layer Design

2.2.1 Historical Background

The Advent of QoS and Real-Time Multimedia

The layered TCP/IP architecture has proved robust, cheap and scalable from its origins in the 1970's. The exponential augmentation in the number of connected hosts is certainly the most significant fact of the first 25 years of the commercial Internet. Indeed, its underlying technologies and applications had not fundamentally changed form the ARPANET times. Two major facts mark however a turning point in the telecommunications sphere in the late 1990s: the emergence of cheap wireless mobile technologies and the explosive demand for real-time broadband multimedia services. Two direct consequences — rapidly understood by Internet service providers and satellite operators wanting to remain competitive — were the emerging demand for differentiated QoS and the promises of seamless and ubiquitous connectivity.

Unfortunately, coping with these market trends is by no means an easy task with the existing infrastructure. TCP/IP networks were designed for "best-effort", connectionless packet-switched operations, and satellites are optimized for unidirectional broadcasting essentially. Financial investments in both the Internet topology and satellites have been so huge over the past decades that introducing something revolutionary to face the situation is not really an option. The main

challenge resides therefore in the *adaptation* of *both* TCP/IP networks *and* satellites to deliver QoS guarantees efficiently at affordable prices.

Inadequacies of Satellites

Most satellite systems used for interactive services delivery inherit their architecture from a broadcast-oriented design, originally intended to provide media contents to a large panel of receivers in a point-to-multipoint network configuration. Efficient data carriage over satellite suffers therefore from the inefficiencies and difficulties of properly mapping variable-length and bursty network layer packets — such as IP datagrams — into fixed-length link-layer entities not initially intended for that. Such operation is classically ensured by adaptation layers (see Figure 2.3) such as MPE [8], ULE [9] and AAL5, network-to-link layer interfaces having a *major* impact on the overall transmission efficiency through their added overhead (protocol control information, integrity checks, padding) and complexity.

Some ways to improve the efficiency of adaptation layers are examined in Chapters 3 and 4 of this thesis.

Inadequacies of the TCP/IP Stack

Some real-time communications requirements conflicting with the TCP/IP architecture are [37][38][39]:

- Fast adaptability to dynamically changing network and traffic conditions
- Good performance for large networks and large numbers of connections
- High effective capacity utilization
- Low overhead in header bits per packet
- Necessary in-order packet delivery

Common problems with TCP/IP stacks such as packet losses and reordering, retransmissions and suboptimal congestion management are not only conflicting with these requirements: they are often amplified by the varying nature of wireless channels.

2.2.2 Shortcomings of Layered Architectures

Generally speaking, layered architectures de-correlate users' needs and lower layers services by definition: such is perhaps the most important criticism of layering. Information is indeed lost during layer-by-layer conversion (which is particularly critical in the satellite scenario where the physical layer imposes particular constraints), resulting in various inefficiencies and redundancies.

When analyzing in detail the different levels of the protocol stack, for every layer it is possible to identify issues where layering is affecting performance, especially in the satellite case. The following

list reviews the layers of the satellite model presented in Figure 2.3, and describes some of these inefficiencies.

Application

- Error-tolerant applications such as VoIP suffer from systematic discarding of packets at the lower layers, although most of them would be glad to receive faulty data instead of going through retransmissions.
- Application-level contents are generated regardless of the capacity of the system to deal with them. For instance, buffers overflow can be experienced for real-time applications such as streaming during bad channel conditions.
- Dynamics of real-time applications controlled by users' inputs often require fast re-configuration capabilities from the system, but lower layers often require slow dynamics to ensure e.g. proper data interleaving. Large zapping times for mobile TV in DVB-H [40] and DVB-SH [41][42] are a good example of this problem.

Transport

- Erroneous packets drops (especially in the link and adaptation layers) are classically seen by TCP as indications of congestion in the network [43]. TCP therefore reacts by decreasing its transmission window and by resuming transmission in a conservative way, in order to avoid network overflow. TCP's reaction is completely unadapted to the real cause of the problem.
- Reductions in TCP's sending window (either caused by packet drops or by actual congestion) may need several RTTs to recover, especially when multiple losses occur that may cause a TCP timeout [44][39], meaning seconds in the satellite case.
- In addition to its congestion algorithms, TCP's slow-start is known to be inefficient when used over a network path with large DBP [45]. TCP's startup performance could be therefore improved with explicit information about the current available capacity of the connection path [46], obtained from both the link and the physical layers. Several wireless and satellite-friendly TCP flavors tackle these issues as well [47].
- RFC 3819 points out that there is no relation between subnetwork connections and any connections that may exist at higher layers such as TCP. An important consequence is that TCP timeouts are not related whatsoever to ARQ dynamics, which leads to undesired situations. Consider for instance that ARQ is retransmitting (parts of) a packet, making the link latency momentarily increase. Since TCP bases its retransmission timeout on prior measurements of total end-to-end latency, including that of the link in question, this sudden increase in latency may trigger an unnecessary retransmission by TCP of a packet that the link layer is still retransmitting. Such spurious end-to-end retransmissions generate unnecessary load and reduce end-to-end throughput. As a result, the link layer may even have multiple copies of the same packet in the same link queue at the same time. [48].

Network

- Mobility affects network performance by inducing delays and reduced throughputs during hand-offs. For this reason [45] and [49] point out that contrary to current network design practices, it seems better not to hide mobility events (e.g. hand-off initiation and completion) to the network layer.

- IP requires that every packet carries routing and protocol information in headers so that each of them has an independent life in the network. Although this is a deliberated design choice, current headers are so large that they end up consuming a very important bandwidth portion. Note in addition that the upcoming of IPv6 and the increased use of tunneling mechanisms e.g. for mobility mark a tendency towards the aggravation of this issue.

Adaptation

- The adaptation layer fragments (resp. reassemblies) PDUs into (resp. from) several link layer frames. This requires overhead, padding, and increases PDU vulnerability given that a single erroneous frame causes the loss of an entire PDU under current designs.

- Cyclic Redundancy Checks (CRCs) used in adaptation layers can be somewhat redundant. Although useful for detecting flawed SNDU reassembly, their usefulness can be questioned under good channel conditions and correct FEC operation.

- Currently used fixed-length frames such as ATM cells or MPEG2 packets [50] are not well adapted to the bursty and variable-length nature of IP datagrams. Blind PDU encapsulation into link layer frames without taking traffic characteristics into account leads to excessive delays or to excessive padding, hampering transmission efficiency.

Link[3]

- Satellite and wireless channels offer much more possibilities for non-colliding and opportunistic channel access than wired links [38]. Constant bandwidth allocation inspired from wired networks is clearly an inefficient option on satellite links. Although standard Dynamic Assignment Multiple Access (DAMA) schemes in DVB-RCS [51] guarantee an optimal network utilization, they can also impact TCP end-to-end performance if QoS considerations are not properly taken into account. For instance, the preliminary signaling exchange in the request/allocation process introduces additional delay contributions to the RTT that may not be compatible with some QoS classes [52].

- Most deployed link-layer schemes are still dealing with frames in a first-in-first-out basis, since no QoS information is mapped into link-level containers. This very important point has received a great deal of attention in the last years, both from standardization bodies and researchers (see Section 2.3).

Physical

- Given that this layer is responsible for error-free transmission, knowledge related to Channel State Information (CSI) is crucial to handle error control efficiently. Unfortunately, most deployed physical layer schemes use over-robust Constant Coding and Modulation (CCM), coping with worst case channel conditions by ways of safety margins that only prove useful a very limited amount of time. This lack of adaptivity provokes resource waste not only during good link conditions, but also under bad channel situations where receivers unable to decode the signal still get capacity allocation.

[3] The real boundaries between the link and the physical layers is blurry. The inefficiencies listed here may well be imputed to any of them.

- Similarly, physical data transmission does not discriminate flows requiring different application-level QoS, entailing poor service guarantees.
- Physical layers do not provide tools to achieve unequal data protection in a single flow, even though some payloads are clearly more sensitive to errors than others.
- Energy-related issues are of the foremost importance for autonomous and cheap terminals. Layered architectures do not introduce any policy for energy-aware functioning or power-saving policies, considering physical layers as simple bit pipes always on. Researchers on the area of sensor networks have particularly been active in addressing this issue [53].

Regarding these last considerations, note that they are particularly true for physical layers inherited from legacy standards. The advent of sophisticated adaptivity techniques (e.g. FMT) and their natural adoption in next generation standards is rapidly changing that.

2.2.3 Cross-Layer Definitions

We suggest here two definitions for CLD. The first one is somewhat theoretical and abstract, and the second provides a more pragmatic insight on what CLD is.

A First Definition

Cross-Layer can be elegantly defined as *communications system and protocol design by the violation of a reference layered communication architecture* [38]. In our case, that reference layered architecture is the satellite model of Figure 2.3.

A small example taken from [38] quickly illustrates this: take a hypothetical three-layer model with the layers denoted L1 (lowest), L2 (middle) and L3 (highest). In this architecture, there is no interface between L1 and L3. One could, however, design an L3 protocol that needs L1 to pass a parameter to L3 at runtime. This calls for a new interface, and hence violates the architecture. Alternatively, one could view L2 and L1 as a single layer, and design a joint protocol for this "super layer". Or one could design the protocol at L3, keeping in mind the processing done in L1, again giving up the luxury of designing the protocols at the different layers independently. Note that these violations clearly undermine the significance of the architecture since the architecture no longer represents the actual system. If many of them accumulate over time, the original architecture can completely lose its meaning, having a detrimental impact on the overall system as discussed in the last part of this chapter.

A Second Definition: Where Communities Intersect

Intuitively, successful CLD calls for thorough understanding of the various aspects regarding communications systems at the different levels of the protocol stack. Historically, this knowledge has been divided among different communities who have often focused to solve "their specific problem" the best possible way, without a federative holistic direction [37][54][55]. Roughly speaking, these

communities and the issues they have historically dealt with can be summarized as follows:

Signal Processing

- Increasing spectral efficiency in bits/s/Hz
- Reducing the BER
- Reducing the transmission energy
- Designing multi-access algorithms

Wireless Networking

- Defining network protocols
- Defining efficient routing algorithms
- Dealing with scalability issues
- Developing Operations And Maintenance (OAM) tools

Information Theory

- Developing capacity limits
- Defining efficient source and channel coding algorithms

According to [37], *CLD is a communications design approach that combines the resources available in the aforementioned communities, in order to allow for highly adaptive and QoS-efficient links by sharing state information between different processes of modules in the system.* A pictographic translation of this definition is shown in Figure 2.4, taken from [37].

2.2.4 Cross-Layer Types

There are several ways to implement cross-layer proposals in a communications system. A global taxonomy into four major CLD categories proposed in [38] is presented here, based on the nature of the interactions they require from the layers involved. These categories are: the creation of new interfaces, the merging of adjacent layers, context-aware layer optimization and vertical calibration across layers.

Creation of New Interfaces

Most cross-layer designs require the creation of new interfaces between layers, in order for them to share parameters at runtime. Such CLD information can either flow *upwards* (informing higher layers about underlying network conditions), *downwards* (giving hints on how the application data should be processed) or be *bidirectional* among the involved layers. Upwards information flows

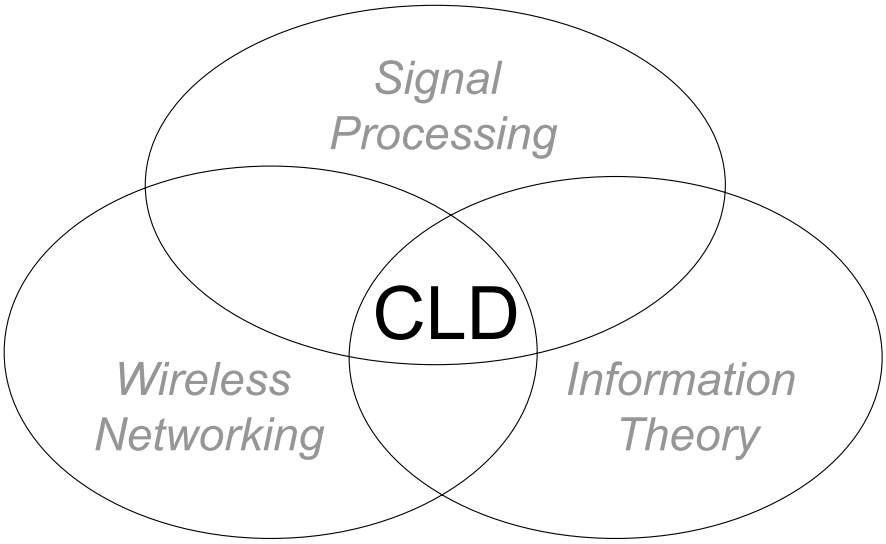


Figure 2.4: CLD: where communities intersect (source: [37]).

through new interfaces are used for example in IETF's Explicit Congestion Notification (ECN) and DVB-S2's Adaptive Coding and Modulation (ACM), both analyzed later. A downwards CLD information flow would for instance translate application-level QoS into link layer parameters, allowing lower layers to treat packets from delay-sensitive applications with priority. Bidirectional collaboration between layers implies not one, but two new interfaces, and may involve iterative loops between the involved layers for instance. Note finally that when dealing with cross-layer exchanges between adjacent layers — beyond the scope of a reference model — a simple approach consists in providing additional primitives to the existing interfaces.

Merging of Adjacent Layers

Merging two layers into a "super layer" is another way to implement a CLD proposal, which does not require the creation of additional interfaces. Architecturally speaking, the new super layer can be interfaced with the rest of the protocol stack by using the interfaces that already exist in the original architecture. Up to now, no explicit proposal has been done in this sense, although many proposals tend already to blur the boundary between some adjacent layers such as L1 and L2.

Context-Aware Layer Optimization

Optimizing a set of layers by adapting their behaviour to a particular network environment without affecting their network interfaces can be seen as a particular form of CLD. For instance, one can design a given layer with a specific channel in mind. In another example, a layer i can be coupled to another layer j. In the latter case, the architectural cost is that it may not be possible to replace one layer without making the corresponding changes to another layer.

Vertical Calibration Across Layers

Finally, vertical calibration refers to adjusting parameters that span multiple layers. This is perhaps the most ambitious form of CLD, given that the performance seen at the level of the application is a function of the parameters of all the layers below. Such calibration can be *static*, e.g. by setting parameters at design time with the optimization of some metric in mind. It can also be done *dynamically* at runtime, emulating a flexible protocol stack that responds to variations in the channel, traffic, and overall network conditions. The former does not create significant implementations issues since the concerned parameters are adjusted from the beginning. However, the latter may bring significant complexity, energy consumption and overhead, in addition to stringent conditions on the retrieval and update process to make sure that the knowledge of state of the stack is current and accurate.

Summary

Figure 2.5 summarizes the four categories of cross-layer designs previously described, taken from [38].

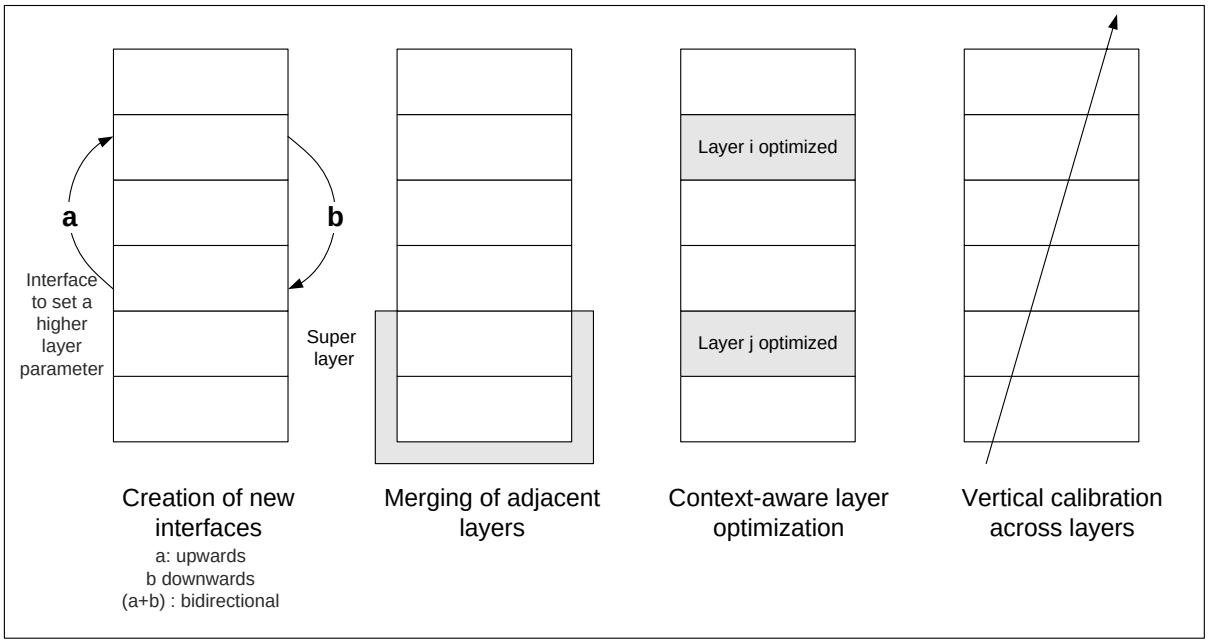


Figure 2.5: Illustrating the 4 kinds of CLD proposals (source: [38]).

2.2.5 Towards a Cross-Layer Architecture

We have just attempted to describe CLD proposals by analyzing the type of interactions they require between layers. Now, possible implementations for such interactions are reviewed.

Intra-host vs. Inter-host CLD

The aforementioned CLD types deal with layers as abstract and unique blocks. However, it is important to keep in mind that in general each layer represents at least two separate protocol entities running at end points separated by a network [56]. This implies that CLD proposals can be further classified into *intra-host* or *inter-host* (often called *explicit* CLD), depending on *where* cross-layer exchanges are done among the layers involved.

- Conceptually, intra-host CLD implies hardware or software modifications affecting a single end point, without any kind of network involvement: layers exchanging information do so inside the concerned device exclusively. Generally speaking, intra-host CLD aims at improving short-term usage experience of a device such as e.g. a mobile phone, by allowing its battery to last longer or to run multiple applications simultaneously and smoothly.
- On the other hand, inter-host (or explicit) CLD involves protocol dialogs across the network. It aims at improving long-term usage experience by focusing on network and link efficiency, rather than on short-term considerations affecting a single node. Note that inter-host CLD is inherently more complex than intra-host CLD, since special attention should be paid to the interaction between CLD messages and middle-boxes or routers in the path as described in Section 2.4.

Direct Layers Communication

A straightforward way to allow runtime information exchanges between layers is to allow them to communicate with each other directly. Practically speaking, this means making the variables at one layer visible to the other layer by the definition of proper "request" and "retrieve" primitives. By contrast, under a strict layered architecture, every layer manages its own variables which are of no concern to other layers.

There are many ways to establish CLD signaling for direct layers communication, of which [57] provides a compact classification. For our purposes, these categories can essentially be summarized in *in-band* and *out-of-band* types [45].

- *In-band* signaling goes with normal protocol traffic, that is with a protocol control message or data such as headers, both for intra-host and inter-host CLD. The benefit of in-band signaling is that inter-host CLD messages are known to share the same path as normal protocol traffic, and generally use less overhead than a separate message. The disadvantage is that if some routers and middle-boxes drop a packet because of unknown protocol information (e.g. a notification transmitted over an IP option), the accompanying data gets lost as well.
- *Out-of-band* mechanisms for inter-host CLD require a dedicated protocol for signaling. While out-of-band mechanisms save the normal protocol traffic from additional overhead, the transmission of separate messages may be prevented by middle-boxes on the connection path in inter-host CLD. If the message is lost in the network for some reason, there may not be any way for either end of the connection path to know about it. If the out-of-band notification

needs to be matched with a particular flow, the notification message would need to include the IP source and destination address, transport protocol, and source and destination transport protocol ports. Getting and using this information may not be possible in all cases, for example when the transport protocol header is encrypted by IPsec ESP [58]. If the notification needs to be in synchrony with the data flow, a separate out-of-band message may be problematic, because the message may be lost or delayed relative to the data traffic. Out-of-band signaling in intra-host CLD only affects the device implementing the proposal (and not its network environment), for which such CLD proposals will often be protocol-compliant and transparent, without requiring added overhead.

A Shared Database

Another way in which layers can communicate with each other is through a shared database, sometimes referred to as a cross-layer manager or device management entity [49]. In one sense, the common database is somewhat like a new layer, providing the service of information storage and retrieval to all the layers through a dedicated API. The approach relying on a shared database is particularly well suited to vertical calibration across layers.

Summary

Figure 2.7 summarizes the two aforementioned architectures for CLD implementation, and Figure 2.6 illustrates the concepts of intra-host and inter-host CLD.

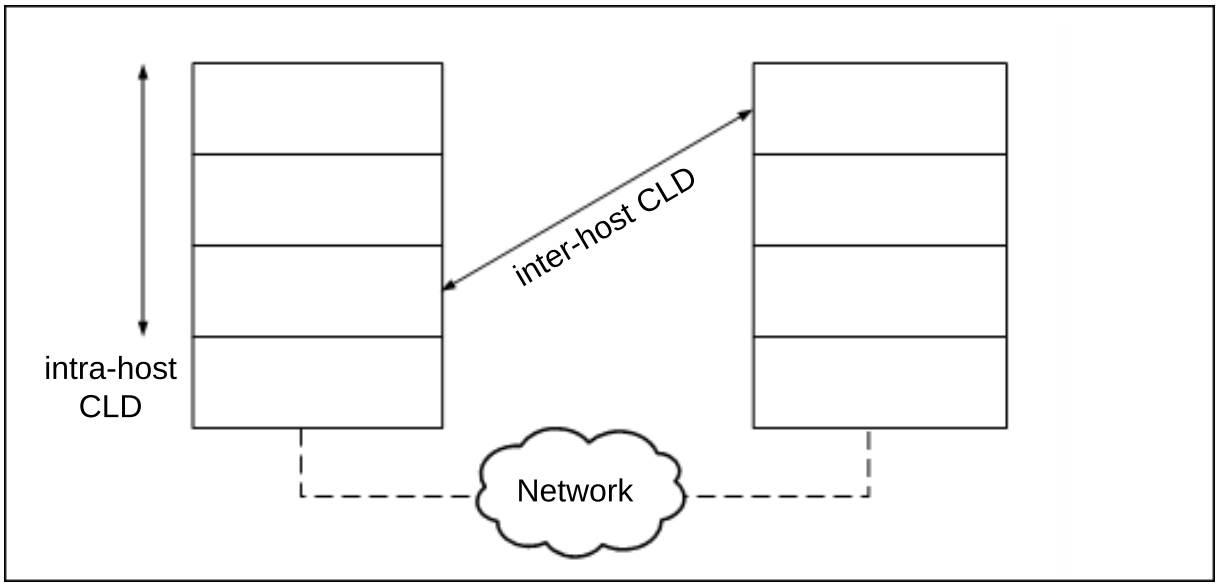


Figure 2.6: Intra-host vs. inter-host CLD.

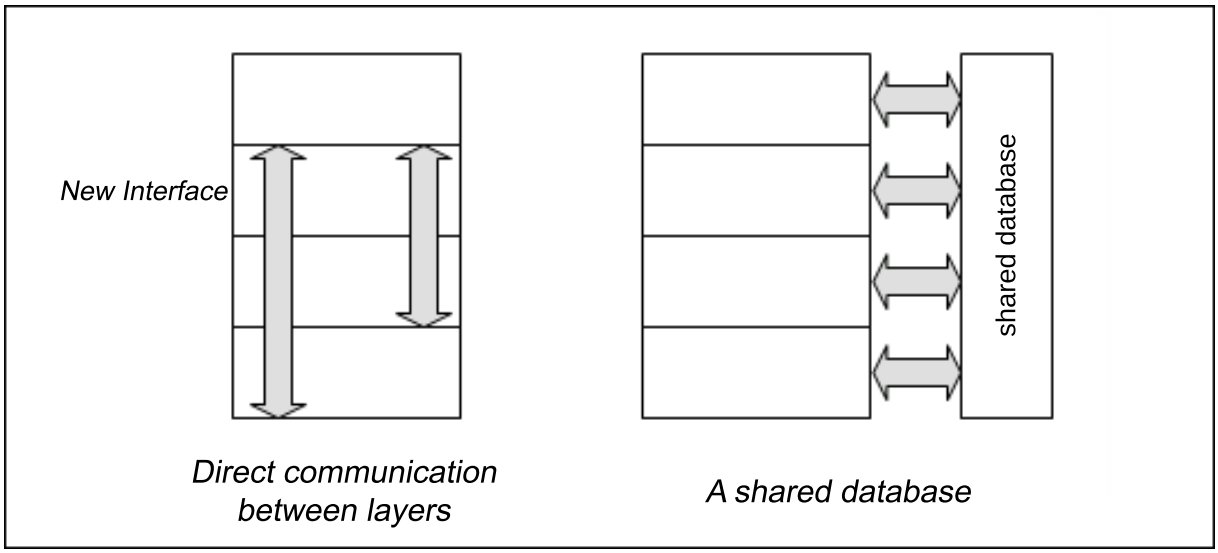


Figure 2.7: CLD architecture proposals (source: [38]).

2.3 Past and Current Cross-Layer Design Proposals: an Overview

Congestion & rate control, resource management and QoS are the three major axes that have received the greatest deal of attention from CLD researchers. Without surprise, these are the points were layered architectures have proven weaker as the analysis of the layered stack showed. The work presented in this thesis relates to CLD proposals for error control, which could be classified in the QoS category. However, the general problematic of error control has important implications in the other two aforementioned categories.

The following lines present an overview of the major existing CLD proposals in these three areas.

2.3.1 CLD Proposals for Congestion and Rate Control

Without surprise, most of the proposals in this area have originated from the network community led by the IETF [45].

Explicit Congestion Notification (ECN)

The first transport layer issue evoked in Section 2.2.2 is perhaps one of the best known issues related to layering. Several CLD proposals allowing transport layer discrimination between packet drops due to errors and congestions in error-prone networks have therefore been proposed in recent years.

Among them, the Explicit Congestion Notification (ECN) [43] is with no doubt the most successful, and most CLD proposals dealing with congestion control are somehow related to it. ECN uses a two-bit field in the IP header to allow routers to indicate congestion in the network before they have

to start dropping packets due to buffer overflow. ECN can be useful even if only a subset of routers implement it on the connection path. There were initial deployment problems with ECN because some routers in the network dropped packets with a non-zero ECN field in the TCP header, but it is believed that today most of these routers have been fixed [45]. ECN is an inter-host in-band mechanism, requiring a more complete L3-L4 interface.

Note also that the use of the ECN field is taken further in an alternative protocol to use the field, called Re-ECN. The protocol aims "to provide sufficient information in each IP datagram to be able to hold senders and whole networks accountable for the congestion they cause downstream, before they cause it" [45].

TCP Quick-Start

In Quick-Start, described by RFC 4782 [59], the sender uses an IP option to request permission from routers to send at a higher rate than the normal congestion control would allow, coping with the second transport layer issue in Section 2.2.2. The RFC specifies the use of Quick-Start for TCP and discusses the challenges such a mechanism needs to address. Quick-Start router algorithms and their configuration are analyzed further in [60], and [61] gives an initial analysis of Quick-Start in wireless environments with vertical hand-offs between different wireless link technologies. Quick-Start is also an inter-host in-band example of CLD requiring a more complete L3-L4 interface.

IP Mobility

The interaction between congestion control and mobility is a very important point to be addressed in future wireless and satellite networks, e.g. for covering fast moving trains and aircrafts. Seamless and transparent mobility management requires keeping addresses and port bindings active, and smooth hand-offs to be dealt with in the network. However, as mentioned in Section 2.2.2, such events are very bad handled by layered architectures and especially by TCP's congestion control procedures. This, and many other issues related IP mobility management are addressed in at least 4 different IETF working groups, namely mip4, mext, mipshop and netlmm.

Other Proposals

The following experimental proposals also address the congestion and rate control problem with a cross-layer approach.

The Variable-structure congestion Control Protocol (VCP) is another congestion-control proposal using explicit feedback from routers. VCP leverages the ECN field to let routers indicate their load information. Based on the VCP bits, a TCP sender could apply either Multiplicative Increase, Additive Increase, or Multiplicative Decrease of the congestion window. VCP is an inter-host, in-band example of CLD requiring a more complete L3-L4 interface[4].

[4] According to [45], VCP does not provide a mechanism for checking that all routers have understood and processed the notification. It is possible that VCP allows Multiplicative Increase even if there are fairly loaded routers on the

The Explicit Loss Notification (ELN) [62] and the Explicit Transport Error Notification (ETEN) [63] are other in-band mechanisms that have been proposed for dealing with corruption-based packet losses in wireless and satellite networks. ETEN is also an inter-host in-band mechanism.

The experimental Explicit Control Protocol (XCP) [64][65] is a proposal for a full-fledged congestion control protocol involving interaction between routers and end-hosts. XCP uses a separate congestion header between IP and the transport protocols sent with the data: it is an inter-host in-band mechanism.

Cross-layer TCP Selective Acknowledgements (CSACK) [66] is based on an out-of-band intra-host mechanism allowing the link layer to send congestion notifications to the transport layer, in a scenario using Performance Enhancing Proxies (PEP) that maintain TCP split-connections.

2.3.2 CLD Proposals for Quality of Service

It can be argued with reason that any enhancement to the system will have an impact on the perceived QoS by the final user. For this reason, many of the proposals presented in the following lines could be classified elsewhere. We have chosen to include here those mechanisms that have a direct impact on users' experience without really affecting the underlying network. Agreement of QoS promises in terms of delay, jitter and achieved BER are the metrics used here.

Partially Reliable Delivery Mechanisms

Partially reliable delivery mechanisms try to cope with the first application layer issue stated in Section 2.2.2. Since real-time streaming traffic may use codecs that are error-resilient, it is possible to receive data with (some) bit errors within the packet payload. This suggests that application-layer packets can be divided into "sensitive" and "insensitive" parts with differentiated error control capabilities. Errors in the sensitive part cause a packet to be discarded whereas packets with errors in the insensitive part are delivered, leaving the final decisions on its use to the application layer.

The in-band inter-host UDP-Lite transport protocol [67] standardized by the IETF in RFC 3828 [68] allows the coverage of UDP checksums to be modified. This directly specifies the limit in the carried packet between its sensitive and its insensitive parts. Complementary attempts to adapt IPsec and other tunneling-using mechanisms to UDP-Lite are described in [69].

Similarly, the Multi-Protocol Header Protection [3][70] proposes a multi-layer FEC scheme devised to cover the cumulated header of every packet, adding locally redundancy bits that protect header fields from being corrupted.

Unequal error protection codes and partial order connection services [71][72] can also be considered as partially reliable delivery mechanisms. However, rather than differentiating among sensitive and non-sensitive packet parts, these techniques make that distinction at packet (or frame) level. For instance, [73] applies these concepts to MPEG4 video streams, where protection classes are

connection path that do not support the mechanism. Therefore, VCP would be an invalid mechanism to be deployed in the Internet.

naturally derived from the different importance of I,B and P frame types in the flow. Another application of unequal error protection techniques for progressively encoded multimedia sources via RTP is dealt with in [74].

Rationalization of Error Control Techniques

Exploring new error control mechanisms by means of CLD techniques is the purpose of this thesis. In this context, two mechanisms have been developed during this PhD work.

The first of them is described in Chapter 3. It analyzes the inefficiencies of concatenating physical-level FEC and adaptation-level CRCs without allowing them to share any knowledge on the decoding progress. It is shown that in most practical cases, an intra-host cross-layer mechanism allowing the adaptation layer to keep track of FEC's decoding status can make CRC checks redundant, thus saving their associated overhead. Both in-band and out-of-band implementations of this CLD proposal can be devised.

Next, the intra-host out-of-band Header Redundancy Assisted Cross-Layer Error Suppression (HERACLES) framework presented in Chapter 5 uses upper layers redundancy between headers to correct errors at physical level with excellent accuracy. It achieves powerful error control and provides delineation capabilities for packet synchronization even under extreme noisy conditions.

So far, the author has no knowledge of other cross-layer techniques focusing on error control for satellite links.

Scheduling Proposals

Packet scheduling in the lower layers according to upper layers criteria (e.g. DiffServ) is at the very heart of any QoS-capable service; its absence is the foremost important weakness of classical link layers. The recent definition of the DVB-S2 standard by ETSI has motivated intense research in this domain, given its intrinsic adaptive capabilities.

Among the latest proposals in this sense, [75] presents a cross-layer technique for the design of the forward link packet scheduler that introduces fairness as a tunable parameter for unicast services by DVB-S2. Its approach makes it possible to adapt dynamically the scheduler behaviour depending on the channel conditions in order to guarantee fairness. The proposed algorithm also supports differentiation of services that complies with the requirements for implementing QoS.

Other CLD proposals for scheduling can be found in [54] and [76].

Other Proposals

The following lines present other cross-layer initiatives that affect QoS in different ways.

The Resource ReSerVation Protocol (RSVP) uses separate out-of-band messages on top of IPv4

or IPv6 to make QoS signaling [77]. The data sender sends a RSVP "Path" message to the data receiver that includes a Router Alert IP option telling the routers on the path to investigate the RSVP message contents closer. Each router adds its IP address to the message to enable routing of the Reservation (Resv) messages sent in the reverse direction to visit exactly the same routers on the reverse path to the data sender. The Resv message does not use the Router Alert option, but is rather explicitly routed on a hop-by-hop basis between the network routers using the state established earlier. In addition to the Path and Resv messages, RSVP has a few other message types delivered on a hop-by-hop basis. RSVP is clearly an out-of-band inter-host CLD mechanism.

Recently the IETF has specified a NSIS (Next Steps in Signaling) framework to handle signaling in the Internet. The Generic Internet Signaling Transport (GIST) protocol has been specified to transport application-specific signaling messages over the Internet [78]. GIST messages are transferred using TCP or UDP as the transport protocol, depending on whether a reliable connection-oriented service or a connectionless service is desired. GIST has some common characteristics with RSVP: it uses a Router Alert option to wake up the GIST-aware routers along the path, and for further signaling, explicit hop-by-hop routing can be applied using the state established at routers. Like RSVP, also GIST is an out-of-band inter-host scheme.

2.3.3 CLD Proposals for Resource Management

Resource management is the third important axis for current satellite cross-layer optimization.

Fade Mitigation Techniques (FMT)

Link quality degrades significantly during adverse weather conditions, especially in frequency bands above 10 to 15 GHz. Satellite systems have therefore implemented high system static margins, in order to insure a minimum outage duration of the service for a given objective of link availability. Fade Mitigation Techniques (FMT) allow systems with rather small static or dynamic margins to be designed, while overcoming *in real time* most fading events [4][79]. Dynamic adaptation requires CSI to be estimated or measured in order to be used at different levels of the protocol stack.

Among those techniques, Adaptive Coding and Modulation (ACM) is of high interest as it allows the performance of *individual links* to be significantly optimized [80], especially for interactive services [81]. ACM consists in tuning both coding and modulation parameters of the physical layer, in order for individual receiver characteristics to be adapted to propagation conditions and service requirements for the given link. De facto integrated in the new DVB-S2 standard, ACM is a long-awaited breakthrough in satellite communications that has motivated many works and interesting ideas in the satellite field [82]. In particular, the issues raised when encapsulating IP datagrams over ACM-controlled frames are dealt with in detail in Chapter 4. The framework developed in this context eventually contributed to the definition of the Generic Stream Encapsulation protocol [25][28].

Bandwidth Allocation Techniques

Proposals for cross-layer bandwidth allocation schemes coping with the limitations of current methods such as DAMA (see Section 2.1.3) have flourished in the recent years, given its crucial importance not only in the satellite field, but also for cellular and wireless LANs [83][84].

Most of these proposals aim at establishing bandwidth allocation strategies driven either by TCP's internal state, local channel conditions or both. For instance, in [52] resource requests are synchronized with the TCP congestion window trend in order to assign or remove dynamically capacity. Reference [85] makes stronger use of CSI knowledge in order to proceed to radio resources allocation, making close ties with the aforementioned FMT.

Reference [86] goes a step further by analyzing a Packet Reservation Multiple Access with Hindering States (PRMA-HS) whose parameters depend both on the observed CSI and on the service type dictated by the application layer.

2.4 Challenges and Open Issues

So far, CLD has demonstrated a true potential for performance enhancement. However, it is still a young research area where many issues and challenges are to be addressed. Some have even pointed at the risks of developing an overly confidence on CLD, running at cross purposes with sound and long term architectural principles that have proven good so far. What can be therefore expected from CLD in the coming years?

The following paragraphs describe some of the major challenges and open issues that CLD researchers face today, and conclude with a short discussion on its perspectives.

2.4.1 Implementation Challenges

Intra-host CLD, as opposed to inter-host CLD, only affects the internal functioning of a device where it is implemented. The expected issues with intra-host CLD are therefore added complexity, memory/processing requirements, and extra energy consumption. The potential benefits of intra-host cross-layer optimizations have therefore to be analyzed with these metrics in mind.

Requirements for inter-host CLD differ radically than for intra-host CLD, due to the involvement of the network in the transport of cross-layer messages. Classical issues such as security and network stability have therefore to be addressed. The following excerpts from [45] illustrate some important challenges faced by inter-host CLD.

Security Issues

When implementing inter-host CLD mechanisms, a certain number of security issues arise. Of course, assuming that the use of IPsec will solve them is all the more wrong since IPsec has never

been intended to cope with security issues beyond the strict TCP/IP framework. Furthermore, many cross-layer proposals may be incompatible with IPsec, like UDP-Lite for instance [67][69].

A cross-layer signaling protocol needs protective measures that are strong enough to make attacks on the protocol difficult and reasonably unprofitable. At the same time, if an otherwise light-weight protocol has heavy-weight security mechanisms, the cost of the security procedures may outweigh the possible benefits of the protocol.

For in-band mechanisms that use reserved header bits or IP options, the receiver of the packet can be expected to check that the IP addresses and transport ports match the existing connection, and that the sequence numbers in the packet belong to the currently valid window. Therefore, blind attacks generated outside the packet transmission path have a reasonably low probability of succeeding. However, an attacker on a connection path that is able to read the transport and IP headers has a good chance of causing harm to a connection, particularly if the packet contains additional explicit information about the connection, for example in an IP option. IPsec can protect the transport header, but does not protect a mutable IP option that can be modified by routers along the path.

Out-of-band messages do not necessarily include the additional context from the transport protocol, so they can be an easier target for blind attackers. If a transport protocol context exists, for example when the message is triggered by a data packet, the sender of the out-of-band signaling message can include the transport header from a recent data packet with the message to authorize the message based on the "proof" that the message has come from the right source. In principle it cannot be assured that an out-of-band message uses the same path as the data traffic, although it can be assumed to be a common case.

IP Tunnels

IP tunnels are a challenge for cross-layer notification protocols that require routers participation, because tunnels isolate the original IP header inside an outer header. A tunnel protocol could copy the important cross-layer notification data to the outer header at the tunnel ingress so that the routers along the tunnel path can process the information, and then at the tunnel egress copy the possibly changed cross-layer data back to the inner header. For IPsec tunnels there is a special consideration whether exposing the cross-layer data in the outer header is a violation of the security policy. It is possible that some additional cross-layer information on the outer header makes it possible for an intruder to make additional conclusions about the nature of the data that is being transferred inside the IPsec tunnel.

Because the interaction of congestion control and mobility has been one of the key motivations for advanced cross-layer interactions (see Section 2.3) it is worth noting that one of the most common mobility mechanisms, Mobile IPv4, is based on the use of IP tunneling [87]. When a mobile host is not at its home location, the Mobile IPv4 home agent receives the packets on behalf of the mobile host, and forwards them to the care-of-address of the mobile host in an IP tunnel. There can also be deployments with several layers of tunneling, for example when IPsec is used together with Mobile IPv4. IP tunnels are a particular challenge for mechanisms involving all routers in the path, because currently there is no known guaranteed way to check that the CLD notification has indeed been processed by all routers when there is an IP tunnel on the connection path. The Quick-Start

specification includes a thorough discussion of problems with IP tunnels [59].

Non-Conformant Routers and Middleboxes

The presence of routers, middleboxes or drivers that drop packets containing unknown options (e.g. IP options) would be a major obstacle to any cross-layer mechanisms that depended on the use of such options. With in-band mechanisms this would also prevent delivery of the data in the packets, while with out-of-band mechanisms data transfer would not be directly affected. For schemes that typically need to modify the IP header, this is a particularly important problem.

Processing Efficiency

Packets with IP options are assumed to take the slow-path processing path in most routers, as opposed to the optimized fast-path. If the use of IP options or other mechanisms requiring router attention gained in popularity, the impact on the processing efficiency of routers would have to be considered. In the Quick-Start proposal, it is assumed that Quick-Start capable routers would rate-limit the number of Quick-Start requests that are processed, to preserve router efficiency and to protect against possible attacks on the routers themselves.

2.4.2 Open Issues

This section presents a literature compilation of the major open issues related to CLD [38].

Coexistence and Interoperability of CLD Proposals

An important question to be answered is how different cross-layer design proposals can coexist with one another, both within a system and between other systems. By definition, cross-layer enhancements span two or more protocol layers, with the result that state in one layer can be coupled to state in another entity at a different protocol layer. An attempt by two methods to modify the same state could have a serious and unpredictable negative impact on performance and on system stability. Regarding this particular issue, RFC 4907 "Architectural Implications of Link Indications" [88] is of special interest. It is perhaps the most up-to-date document the IETF has produced on cross-layer mechanisms, and its examples of poorly coexisting cross-layer proposals are quite illustrative.

Determination of CLD Applicability over a Functional Domain

Reference [89] describes an example that illustrates how a cross-layer design involving an iterative optimization of throughput and power leads to a loss in performance *under a certain pathological network condition*. The underlying idea is that cross-layer proposals designers need to establish

the network conditions under which their design proposals should and should not be used. Given that channel dynamics are often faster than those driving upper layers reconfigurations, this point is particularly important for CLD proposals requiring CSI inputs. For this reason, efficient mechanisms to make a timely and accurate assessment of the state of the network will certainly need to be built into the stack.

Interfaces Standardization

A key point for ensuring long-term viability — and possible interoperability — of cross-layer proposals is the standardization of the interfaces required to achieve cross-layer communication (see Section 2.2.4). ETSI's BSM architecture [36] provides a first step in this direction, through the definition of a standard interface between those upper protocols and procedures in a satellite system that provide IP-based internetworking, and all the underlying satellite-dependent functions that affect the final waveform. Reference [38] points out that addressing this challenge requires assessing the performance cost of every implementation. In particular, it stresses the importance of analyzing the impact in terms of delays and overhead in the retrieval/updating of information on protocol performance, and hence of the complexity of these interfaces.

Better Exploitation of Wireless Media Capabilities

In wired networks the role of the lower layers has been rather small: sending and receiving packets when required to do so from the higher layers with intelligence provided at or above the network layer. Today, however, state-of-the-art physical radio layers concentrate so many sophisticated signal processing functions — such as modulation, coding, interleaving, scrambling and so forth — that they can be assimilated to an intricate series of sub-layers themselves. Without surprise, these tools combined with the inherent nature of wireless media should allow them to play a bigger role in wireless and satellite networks. In particular, cross-layer methods could allow multimedia applications to use the channel in an opportunistic manner.

2.4.3 Discussion

In a turning point in communications history where wireless networks and satellite links are becoming more and more used on a daily basis, it is important to be aware of the possible risks than an excess of confidence on CLD can bring. To close this introductory chapter, the following lines discuss some general weaknesses and strengths of CLD.

Exercising Appropriate Caution with CLD

While *ad-hoc* performance optimizations can bring short-term gains, sound architectural principles are usually based on longer-term considerations. It is therefore difficult to compare the achieved benefits of a given CLD proposal against the negative effects it may have on overall architecture.

Generally speaking, architecture pertains to modularity, standardization and long-scale deployment of interoperable subsystems that can be changed or upgraded without affecting the whole system.

Hence, the first obvious concern with CLD is that once layering is broken, the luxury of designing a protocol — or even an application — in isolation is lost. The effect of any design choice on the whole system is therefore to be considered carefully. What RFC 3439 [33] calls the "Amplification Principle" — popularly known as the "butterfly effect" — particularly applies to complex and heterogeneous systems such as modern wireless and satellite networks. In clear, even the slightest undesired interaction with a remote, seemingly unrelated part of the stack has the potential to generate huge consequences affecting performance and even destabilizing the system.

A well known engineering problem is the effect of undesired coupling between subsystems. As the underlying system grows larger, interdependence risks increase as well. Up to a certain point, undesired coupling may take over if no proper measures have been taken against it, affecting the whole system. Cross-layer designs can also create loops, and it is well known from control theory that in these cases stability becomes a major issue. Only intensive testing of CLD proposals can throw a light on coupling risks, most of which can not be easily foreseen.

Tight coupling also means that systems have less flexibility in recovering from failure states, rising the paramount issue of robustness. The bottom line is that the inherent coupling brought by CLD proposals increases complexity, which in turn is often likely to increase unpredicted failure states.

Referring to sound software engineering principles, RFC 1925 [90] states with a touch of humor: *"it is always possible to agglutinate multiple separate problems into a single complex interdependent solution. In most cases this is a bad idea"*. Code longevity, upkeep and re-use depends on defining clearly separated tasks as classical layered architectures have always allowed. Hard-to-maintain code or systems needing to be updated upon every single modification mean higher development time and financial costs, something regarded by the end user as a lower performance value [89].

Successes of Cross-Layer Design

The undeniable success of the Internet is in part related to the soundness of its architectural baselines, captured by the TCP/IP model. The previous section outlined the importance of modularity brought by solid architectural bases, which entail subsystems reuse and interoperability. There is however the ever present desire, and perhaps the need to optimize the existing systems. After all, architectures are guidelines, not rules carved in marble. In particular, the layered architectures presented in Section 2.1.2 *do* offer the possibilities to optimize end-to-end performance and to offer richer services as the various examples of Section 2.3 showed. But that is far from being all.

Congestion & rate control, resource management and QoS provisioning are the areas where most cross-layer mechanisms have been proposed. Without surprise, they have taken advantage of the new possibilities brought by advanced technology and by the broad possibilities offered by wireless media, where the notion of "link" — as opposite to wired nets — is non-existent. Researchers in the CLD area have shown that transport, network and link layers could, and should have larger interactions in wireless networks than they currently have. They have shown that in wireless communications, the borders between the link and the physical layer are extremely blurry. They have also highlighted the importance of delivering precise and accurate CSI to the whole system, and

started to develop opportunistic and channel-aware protocols that violate on purpose the premises of layered stacks.

The truth is that the huge success of the layered model for wired networks has had so great influence in the way network researchers think, and the financial investments in this area have been so big, that the model has imposed itself as the *default* architecture for wireless and satellite networks as well. However, as [89] points out, it is not at all obvious that this architecture is a priori appropriate for wireless networks. What if what we call "cross-layer design" represents some hints of what a new architecture for wireless — including satellites — link should be? What if all these flourishing CLD proposals were just pointing at a direction that none of us has yet clearly identified?

Personal Thoughts

Many have ridden the cross-layer wave without taking into account the major risks previously evoked, not at all a bad thing in itself. Most of them have brought up interesting ideas that can be easily implemented. Others have proposed schemes with unlikely chances of real-world large scale deployment, e.g. due the dimension of the changes they require. Nevertheless, they have all opened research paths and shown new possibilities for satellite — and in general, wireless — communications unthought of some years ago.

In the author's opinion, the recently coined term "cross-layer" should not hide the fact that from the beginning of computer communications, engineers have put their energies on creating and maintaining systems *that work*, rather than on building well-oiled abstractions. Historically, abstraction in layers came later as RFC 1958 points it out: *"The Internet and its architecture have grown in evolutionary fashion from modest beginnings, rather than from a Grand Plan"* [32]. People behind the ARPANET faced complex engineering problems with specific constraints and resources, and made choices that resulted in having functions and protocols divided in layers as we know today. In a certain sense, they were "cross-layering" without knowing it, and of course, without really using these words. For instance, RFC 4907 [88] recalls that the use of upward link indications within the Internet architecture has a long history. In response to an attempt to send data to a host that was off-line, the ARPANET link layer protocol provided a "Destination Dead" indication, described in RFC 816: "Fault Isolation and Recovery" [91]. Some ARPANET experiments even included link-aware routing metrics calculations.

A wide array of cross-layer designs should therefore be proposed and debated — even the most eccentric ones — since this is the very same approach that gave birth to the current Internet back in the 1970s. However, and although the engineering situation is similar in many aspects, the external constraints are quite different. We cannot escape from the fact that the solutions we devise will have to be compatible with the Internet Protocol, given the massive investments in IP infrastructure worldwide and the irresistible convergence towards IP. TCP/IP architecture, by its successful design and commercial deployment, casts a long shadow. For this reason, unless accepted and federated by a central coordination — or standardization — entity, we believe that the vast majority of inter-host CLD proposals have very small chances to cross the gap from papers to reality. Only the best ones — those providing an added value to the network without compromising its security, usability and stability — will be recognized and adopted in the long-term.

Intra-host CLD raises much less issues in this aspect, since intra-host CLD respecting network

interfaces may go totally unnoticed by the outer world. It is therefore the author's opinion that the most successful real-world CLD proposals in the next years will be intra-host, small, and that they will happen *below the network layer*, even though they might require upper layer inputs. Designers and engineers have here an incredible number of freedom degrees, with implementations that have the potential to make their products stand up from the others. Having big and revolutionary concepts for the whole stack might be today a bit tardy, the time being favorable for discreet and efficient cross-layer enhancements.

Chapter 3

Cross-Layer Enhancement of Error Control in DVB Adaptation Layers

3.1 Introduction

3.1.1 Foreword

This chapter describes the first part of the work done on DVB adaptation layers throughout this thesis, where no standard adaptation layer had been defined for *efficiently* mapping IP datagrams over DVB-S2. It was motivated by the general will to achieve an efficient adaptation/encapsulation protocol that would take advantage of the enhanced physical layer of DVB-S2, and especially of its stronger FEC scheme based on Low Density Parity Check (LDPC) and Bose-Chaudhuri-Hocquenghem (BCH) codes.

As a first step for this work, a preliminary study of DVB-S' FEC was undertaken [17], focusing on its Reed-Solomon outer code. In a second step, strong structural and algebraic similarities between DVB-S' Reed-Solomon (RS) code and DVB-S2's BCH were identified, which allowed a similar methodology to be applied to DVB-S2. The results achieved were conclusive, and showed that the role of Cyclic Redundancy Checks (CRCs) in the new adaptation layer for DVB-S2 could be safely reduced. This leads to non-negligible bandwidth savings for future DVB-S2 links, quantified in around 10% for each packet around 40 bytes. These datagrams currently represent around 40% of the current traffic in the Internet backbone [92] — a high proportion in itself. Given the explosion of interactive applications relying more and more on small packets exchanges — such as interactive gaming or VoIP — this proportion is expected to rise sharply in the next years, making the results of this study all the more relevant.

The results presented in this chapter led to the publication of 2 papers [22][24] and eventually contributed to the definition of the Generic Stream Encapsulation protocol (GSE) [25][28], detailed in Chapter 4.

3.1.2 Problem Statement and Chapter Outline

DVB satellites used for interactive services delivery inherit their architecture from a broadcast-oriented design, originally intended to provide media contents to a large panel of receivers in a point-to-multipoint network configuration. Efficient data carriage over satellite suffers therefore from the inefficiencies and difficulties of properly mapping network layer packets -such as IP datagrams- into link-layer entities not initially intended for such use. Such operation is classically ensured by *adaptation layers* such as MPE [8], ULE [9] and AAL5, placed between the link and the network layers of satellite stacks (see Figure 2.3). Adaptation layers have a major impact on the overall transmission efficiency through their added overhead (protocol control information, integrity checks, padding) and complexity.

Segmentation And Reassembly (SAR) of network-level datagrams into fragments of sizes supported by link-layer frames is one of the most important tasks done by adaptation layers. During this process, at the transmitter CRC is classically appended to every datagram prior to segmentation, and used at the receiver to check the integrity of the sent datagram upon reassembly. CRCs detect and discard datagrams with one or more fragments corrupted by resilient errors of the satellite channel. The necessity for such mechanism has never been called into question, although the reliability of physical layers and the performances FEC schemes have greatly improved in the last years. Unfortunately, the price to pay for the extra protection of CRCs is double: first, they add complexity to the overall system, and second, they consume a non-negligible part of the available bandwidth.

This chapter intends to assess the real usefulness of CRCs in today's satellite adaptation layers under the lights of enhanced error control and framing techniques, focusing on the DVB-S [1] and DVB-S2 [2] standards. Indeed, the outer block codes of their FEC schemes (Reed-Solomon and BCH, respectively) can provide very accurate error-detection information to the receiver in addition to their correction capabilities, at virtually no cost. After recapitulating some known results on linear block codes, we discuss and justify to which extent a cross-layer optimization of global error control can be achieved over DVB-S satellite links by reducing the role of CRCs.

Next, we focus more precisely on the specific case of DVB-S2. At the very beginning of this work, questioning the role of CRCs was all the more relevant when addressing the IP over DVB-S2 mapping as no standard adaptation layer had been specified — and as several cross-layer mechanisms were likely to be integrated in its definition. In particular, we show that the theoretical framework developed in the DVB-S context can be easily extended to DVB-S2, and that the results obtained under the lights of this approach fully justify the design choices made for DVB-S2's brand new adaptation layer, the Generic Stream Encapsulation protocol (GSE).

3.2 Linear Block Codes and Cyclic Redundancy Checks

Consider a systematic linear (n, k) block code C over $GF(q)$ with minimum distance d_{min} in a discrete memory channel with q-ary error probability ε. Linearity means that the $n - k$ redundancy symbols added to the message are linear functions of the original k information symbols. Suppose that a codeword $\overline{x} = (x_0, x_1, ..., x_{n-1})$ is transmitted and let $\overline{y} = (y_0, y_1, ..., y_{n-1})$ be the

corresponding received vector. Then

$$\overline{y} = \overline{x} + \overline{e} \tag{3.1}$$

where $\overline{e}$ is the *error pattern* caused by the channel noise and "+" is the component-wise addition of vectors with elements in $GF(q)$. In digital communications systems, the analysis and decoding of $\overline{y}$ can be done in three different ways. Those are pure error detection, pure error correction, and combined error correction and detection [93].

3.2.1 Combined Error Correction and Detection

A *correct decoding* occurs when $\overline{y}$ is closer to $\overline{x}$ than to any other codeword of C in the space $GF(q)$, using the Hamming distance $d(\overline{x}, \overline{y})$. The received message $\overline{y}$ is said to be contained in the correcting sphere of radius $t = \lfloor (d_{min} - 1)/2 \rfloor$ centered on $\overline{x}$, where t is the *correction capacity* of C. Simple combinatory considerations show that the probability P_c of *correct decoding* is given by:

$$P_c(C, \varepsilon) = \sum_{i=0}^{t} \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i} \tag{3.2}$$

If the received codeword does not lie in the decoding sphere of $\overline{x}$, a *codeword error* occurs with probability $P_w = 1 - P_c$. This probability is also given by:

$$P_w(C, \varepsilon) = \sum_{i=t+1}^{n} \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i} \tag{3.3}$$

Depending on the error pattern $\overline{e}$, codeword errors take two forms, as shown in Figure 3.1. If $\overline{y}$ lies within the decoding sphere of a codeword $\overline{z}$ with $\overline{z} \neq \overline{x}$, the decoder assumes that the transmitted codeword was $\overline{z}$ and the error is therefore *undetectable*, which occurs with probability P_u. However, if $\overline{y}$ does not lie in any of the correcting spheres of the space $GF(q)$, the decoder cannot associate any valid codeword to the sent message and the error is *detectable*, which happens with probability P_d. What particular output from the FEC decoder is associated with a detectable error, and how this information is later shared with the communication system depends on its implementation, and several important issues arise in relation with this particular point. Naturally, $P_w = P_u + P_d$, with P_u given by [94]:

$$P_u(C, \varepsilon) = \sum_{i=d_{\min}}^{n} A_i \sum_{s=0}^{t} \sum_{l=i-s}^{i+s} N(l, s, i) \cdot p(l) \tag{3.4}$$

where A_i represents the weight distribution of C and the term $N(l, s, i)$ denotes the number of error patterns of weight l that are at Hamming distance s to a specific codeword $\overline{z}$ of weight i (the definition of $N(l, s, i)$ is independent of the choice of $\overline{z}$). The term $p(l)$ denotes the probability of a specific error pattern of weight l. While $p(l)$ accepts a simple form, $N(l, s, i)$ cannot be calculated simply in the general case [94]. However, it will be shown in Sections 3.3 and 3.4 that P_u can be simplified for the particular Reed-Solomon and BCH codes we study here.

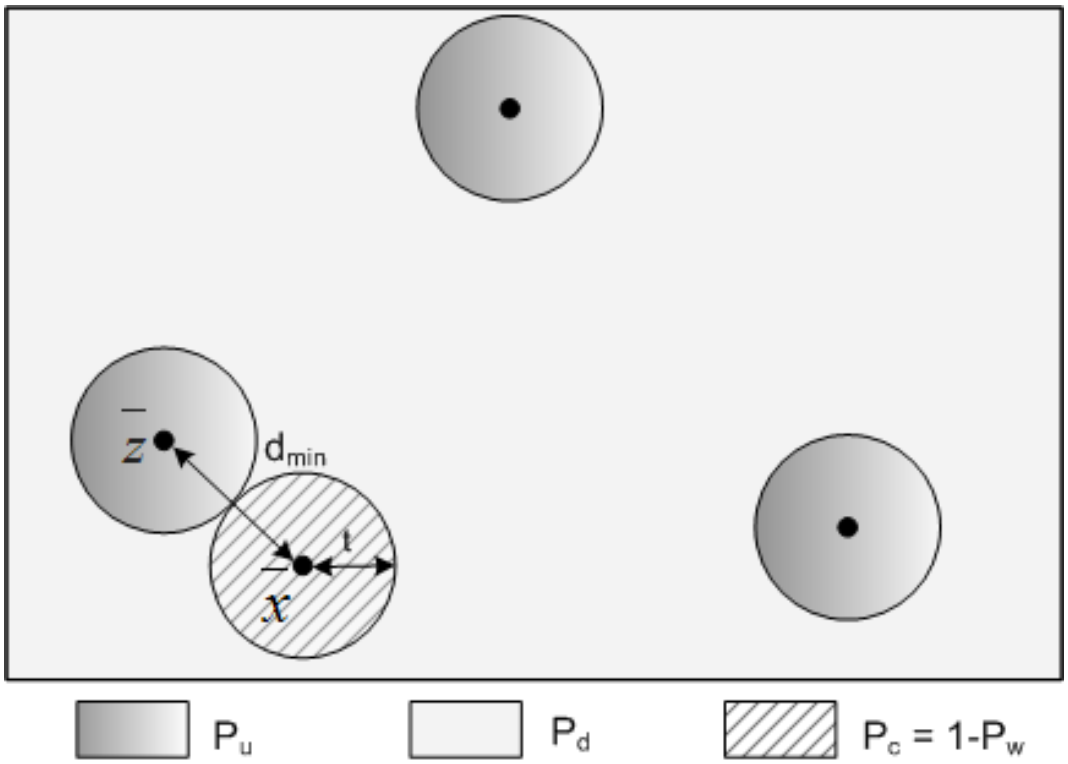


Figure 3.1: Error probabilities and decoding spheres for a linear block code in the space $GF(q)$. $P_c + P_w = 1$ with $P_w = P_u + P_d$ (source: [94]).

3.2.2 Pure Error Detection

Error detection is a particular case of combined correction and detection, in which the decoding spheres are reduced to a singleton, i.e. $t = 0$. The probabilities P_c and P_w of *correct decoding* and of *codeword error* are therefore given by:

$$P_c(C, \varepsilon) = (1 - \varepsilon)^n \tag{3.5}$$

$$P_w(C, \varepsilon) = 1 - (1 - \varepsilon)^n \tag{3.6}$$

The particular fact that the spheres are reduced to a single element greatly reduces the undetectable error probability P_u, since such errors occur only when $\overline{y}$ is identical to a codeword of C different from $\overline{x}$. It has been shown [95] that equation (3.4) can be rewritten for $t = 0$ using the weight distribution A_i of the q^k codewords of C, or the weight distribution B_i of the q^{n-k} codewords of its dual code $C^{\perp}$:

$$P_u(C, \varepsilon) = \sum_{i=1}^{n} A_i \left(\frac{\varepsilon}{q-1}\right)^i (1-\varepsilon)^{n-i} = q^{-(n-k)} \sum_{i=0}^{n} B_i \left(1 - \frac{q\varepsilon}{q-1}\right)^i - (1-\varepsilon)^n \tag{3.7}$$

For C to be good in error detection, this probability should be small for all ε. An upper bound for P_u can be given in the general case of regularly distributed codes [96] in the space $GF(q)$,

assuming that the worst decoding conditions occur when $\varepsilon = (q-1)/q$. For this particular value, every symbol of the q-ary alphabet occurs with equal probability making the channel completely random. Using the second part of equation (3.7),

$$|P_u(C)| = P_u\left(C, \frac{q-1}{q}\right) = q^{-(n-k)} - q^{-n} \leq q^{-(n-k)} \tag{3.8}$$

3.2.3 Pure Error Correction

In pure correction approaches, the decoder always associates $\overline{y}$ with a word of the code, even when the received message does not lie in any of the decoding spheres. Some good examples are convolutional codes, Turbo codes and LDPC codes. However, such a decoding is only efficient when the channel provides soft information on the decoding confidence level, and when the decoding algorithm is able to perform maximum likelihood decoding. The Reed-Solomon or the BCH codes respectively used in DVB-S and DVB-S2 cannot be used in this mode, since there does not exist such computationally tractable algorithms for them.

3.2.4 The Case of Cyclic Redundancy Checks

Cyclic Redundancy Checks used in Ethernet, data storage devices and classical adaptation layers such as AAL5, MPE and ULE are binary ($q = 2$) linear block codes (n, k) used for pure error detection. A CRC_r computed on a k-bit long original Protocol Data Unit (PDU) generates r parity bits, classically appended to the initial message to form a n-bit codeword where $r = n - k$. Since CRCs behave as error detection codes, equation (3.8) applies and:

$$|P_u\left(CRC_r\right)| \leq 2^{-r} \tag{3.9}$$

This makes them excellent error-detection devices (e.g. for $r = 32$, $|P_u(CRC_{32})| \leq 2^{-32} \simeq 10^{-9.6}$), with widespread use in data subnetworks end-to-end checks. Numerical simulations carried on variable-size datagrams sent over a binary symmetric channel show that the 2^{-r} bound is almost always verified for the most widely used CRCs (CRC-4, CRC-8, CRC-16 and CRC-32), or at least, not very badly violated [96]. An example using the generator polynomial $x^{16} + x^{12} + x^5 + 1$ (CRC CCITT-16) is shown in Figure 3.2. Note that P_u does not depend on the size of the protected PDU, and that it is slightly greater than the bound $2^{-16} = 10^{-4.8}$, regardless of the weight of the error pattern $\overline{e}$.

Note finally that classical TCP/IP checksums [97] and most mechanisms relying on hash functions (e.g. MD5 [98]) are not linear schemes.

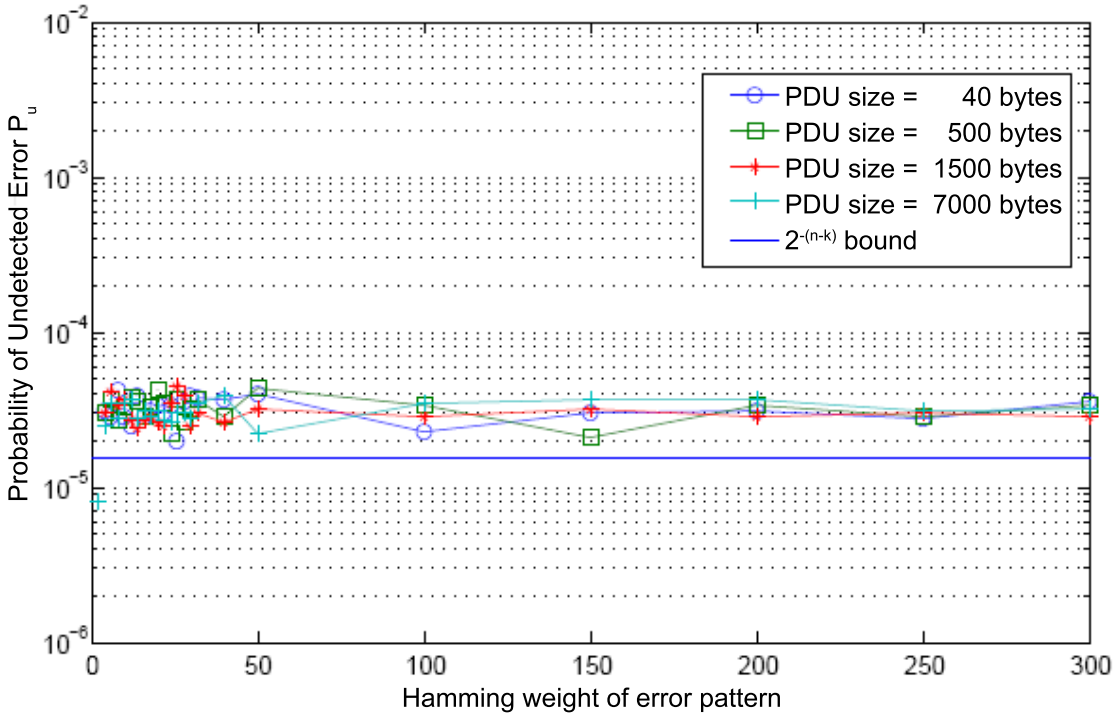


Figure 3.2: Computer simulations for the probability of undetected error P_u for the CCITT-16 cyclic redundancy check.

3.3 FEC-Enhanced Error Control for DVB-S Systems

In the DVB-S standard, an outer Reed-Solomon RS($n = 204, k = 188, t = 8$) code over $GF(2^8)$ (shortened from the original code $n = 255$) and a punctured convolutional code with interleaving are concatenated to achieve *Quasi-Error-Free* (QEF) performances for E_b/N_0 above the operating threshold. The QEF target of the DVB-S standard is defined as "less than an uncorrected error event per hour" corresponding to a frame error rate (MPEG2 level) $FER \leq 10^{-7}$ after FEC decoding. The FEC subsystem of the DVB-S standard is used for combined error detection and correction, and "uncorrected events" stand for *codeword errors*. Although some are *detectable* and some others *undetectable*, as explained in Section 3.2, upper layer CRCs are eventually responsible for dealing indiscriminately with both.

3.3.1 Error Control Management in the DVB-S Adaptation Layer

Every datagram to be sent receives an encapsulation header and a CRC, to form a Sub Network Data Unit (SNDU), whose fragments are carried by different MPEG2 packets forming a Transport Stream [50]. Upon reception, CRCs detect with great accuracy the presence of any wrong data in reassembled SNDUs, and they are therefore used today as the last protection against FEC errors climbing up the upper layers of the protocol stack. When it comes to *undetectable* frame errors, CRCs fulfill their role greatly.

As for *detectable* errors handling, implementations vary. Some produce an erroneous 188-byte frame representative of the final state/iteration of the decoding algorithm, sometimes even containing correctly positioned bits. Other FEC implementations simply replace the packet that could not be decoded with a null packet (e.g. all zeros or all ones) in the binary flow. Note however that in both cases the decoder is aware that the produced output is not a valid codeword and therefore, that there is a detectable error, since this detection is an integral part of the decoding algorithm.

Upon analysis of the incoming flow, CRCs are therefore able to catch both *undetectable* and *detectable* errors coming out from the FEC decoder, no matter their original nature. However, this implies that although the presence of detectable errors is known from the FEC decoder, the CRC has to detect the corresponding series of corrupted SNDUs by himself. In other words, the information generated at the FEC decoder concerning the presence of a detectable error is *never* exploited by the CRC. How often this happens in actual systems is of the greatest importance.

3.3.2 Decoding Error Patterns for the Reed-Solomon Code of DVB-S

Hypotheses

Let's consider $\eta = P_u/P_d$, the ratio of undetectable and detectable erroneous MPEG2 packets (or simply, frames) after FEC decoding. Since MPEG2 packets and classical SNDUs (such as e.g. IP packets) have similar average sizes of few hundreds of bytes, their error rates are in the same magnitude orders. For the sake of simplicity a $1:1$ relation will be supposed to exist between them, so that an MPEG2 error will be said to cause in average one SNDU error, i.e. $FER \simeq PER$.

On the other hand, although the FEC subsystem contains a punctured convolutional code, an interleaver and a RS code, it is assumed that the error-detection capabilities of the overall FEC are those of the RS code, so that the overall η is in fact the one of the RS code. Indeed, the DVB-S specification states that from a functional point of view, the role of the inner convolutional code is to lower the perceived BER at the input of the RS decoder from 10^{-1} or 10^{-2} (actual BER seen at the receiver antenna for a functioning point of E_b/N_0 around 4.5 dB) to 2.10^{-4}.

Finally, it is assumed that the only errors to be dealt with are those encountered at the output of the FEC decoder, since there is no evidence that unexpected hardware/software malfunctioning introduces further errors in the binary flow between the FEC output and the decapsulator input.

Theoretical and Experimental Analysis

Reed-Solomon codes belong to the family of Maximum Distance Separable codes, for which it has been shown that equation (3.4) can be simplified assuming ε is large [95]. Using equation (3.3) the ratio η can be therefore easily found, keeping in mind that $P_w = P_u + P_d$:

$$\eta \approx q^{-(n-k)} \cdot \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \qquad \text{for large } \varepsilon \tag{3.10}$$

In addition, known mathematical properties of RS codes and their weight distribution allow extracting an approximation of η for small values of ε [94]:

$$\eta \approx \frac{1}{t!} \cdot \left(\frac{n - \frac{3}{2}t}{q-1} \right)^t \qquad \text{for small } \varepsilon \tag{3.11}$$

For $q = 2^8 = 256$, $t = 8$ and $n = 255$, η is in the magnitude of 10^{-5} for any ε value using any modulation, meaning that undetectable error events are statistically 10^5 times less frequent than detectable errors under any E_b/N_0 conditions.

Experimentally, a Reed-Solomon code was configured to count the number of times it dealt with detectable error patterns, and a DVB-S link integrating it was modeled with the IT++ library [99]. Extensive simulations run over more than 100 million IP packets encapsulated with MPE allowed to compare this result with the total number of failed CRC checks. They confirmed the theoretical magnitude of η under E_b/N_0 values of 1.6, 1.9 and 2.1 dB, poor link conditions chosen to trigger a large amount of codeword errors upon FEC decoding.

3.3.3 Conclusions and System Enhancement Perspectives

Theoretical and experimental results show that in DVB-S systems, detectable errors at FEC level represent the vast majority of the frame errors encountered after FEC decoding, 10^5 times more frequent than undetectable errors. Therefore, and provided that no further errors affect the binary flow, *99,999% of the failed integrity checks occurring in the adaptation layers can be predicted by the FEC decoder* in average. In other words, CRCs provide original information only 0.001% of the times an integrity check fails in the adaptation layers. Keeping in mind that the QEF target demands $FER = 10^{-7}$ at the output of the FEC decoder for the system to work, this means that CRCs are being really useful only $10^{-5} \times 10^{-7} = 10^{-12}$ of the time the DVB-S link is used. Statistically, this represents an event occurring once every 11 years for a 24 h/day continuous DVB-S transmission.

Under the light of such facts, it seems interesting to set up a cross-layer notification from the FEC decoder to the adaptation layers, in order to optimize or reallocate the resources used today by CRCs. In an in-band implementation, this could consist e.g. in tagging the MPEG2 packets detected as erroneous at the output of the FEC decoder — e.g. by using the Transport Error Indicator (TEI) bit of the MPEG2 header. Such a simple intra-host cross-layer mechanism would allow early discarding of bad SNDUs without the need of a systematic CRC check, while guaranteeing $PER = 10^{-12}$ at the output of the adaptation layer. True, this bound is not as tight as the current level of $10^{-16.9}$ achieved by the current configuration [1], but it is still 100 to 1000 times better than the common best practices defined in RFC 3819 [48]. A step further, the pure suppression of integrity checks in the adaptation layers could lead to the gain of 4 bytes per transmitted packet, meaning up to +10% of bandwidth for small packets, and in a reasonable reduction of the processing load. Figure 3.3 summarizes this in a conceptual way.

[1] $P_u(CRC_{32}) \leq 2^{-32} \simeq 10^{-9.6}$. A CRC_{32} applied over QEF packets $\left(FER = PER = 10^{-7}\right)$ achieves therefore $PER \leq 10^{-16.9}$.

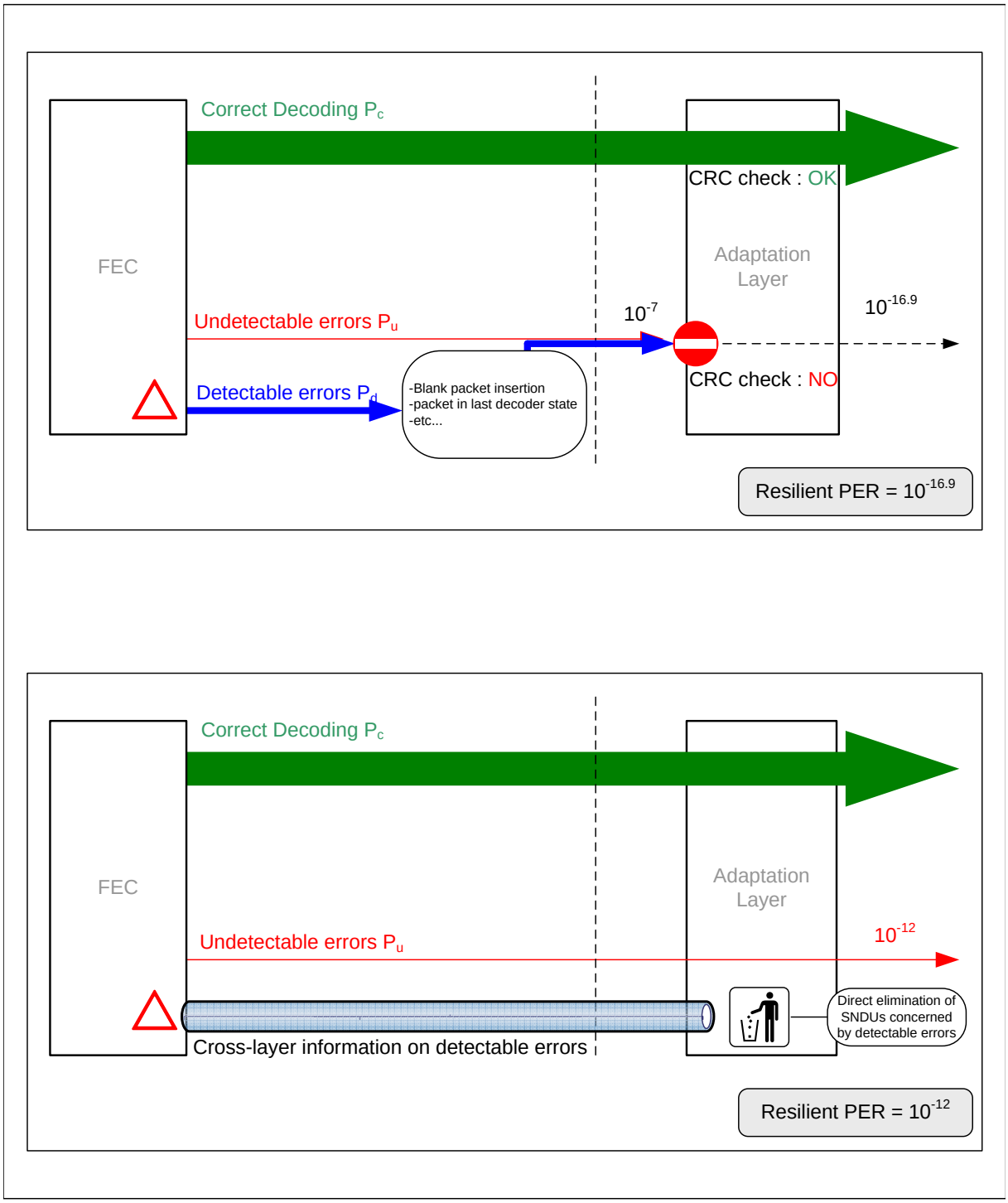


Figure 3.3: Current division of tasks between FEC and CRC in DVB-S adaptation layers (up); proposal for a dedicated cross-layer mechanism enhancing error control (down).

3.4 The Case of DVB-S2

3.4.1 Error Control Management in GSE

The final design choice done for GSE in the area of error control reflects DVB-S2's enhanced FEC capabilities compared to its predecessor. Instead of appending a CRC to every single SNDU following legacy considerations, only fragmented SNDUs (i.e. those placed in frames' edges) are required to carry a CRC in the new adaptation layer. Full SNDUs being carried in the middle of a frame are assumed to be fully protected by the underlying FEC, without the need for any extra protection. The following analyses justify this particular choice, and suggest some possibilities for future cross-layer optimizations as well.

3.4.2 Framing and FEC Considerations

A somewhat detailed description of DVB-S2 can be found in Chapter 4. The following lines give a rapid insight of its most relevant features regarding the analyses of this chapter.

Generic Stream framing

In addition to the classical Transport Streams based on MPEG2, the optional "Generic Streams" framing scheme allows packing network data into a selection of 21 frames of variable payload sizes — 11 long, 10 short — ranging from 0.4 to 7 kB, and offering different payload vs. error protection trade-offs. While broadcast contents are likely to continue using MPEG2 framing, Generic Streams are expected to be privileged carriers for interactive services and data, because of their higher efficiency and flexibility as compared to a MPEG2 mapping using ULE or MPE.

Enhanced LDPC-BCH FEC

Concatenated LDPC and BCH codes are responsible for providing the different error protection levels of the 21 different bearer types, as their overall coding rate is adapted jointly with the modulation scheme according to the radio-link propagation conditions on a frame-by-frame basis. Coded frames (also called FECFRAMEs) are then modulated with one of 4 available modulation schemes (QPSK, 8PSK, 16APSK and 32APSK) defining a wide range of spectral efficiency vs. error protection levels, that can be dynamically allocated for every receiver by an adaptive feedback control loop. Note finally that the overall scheme of the new standard is more powerful than its predecessor, since only 0.4 to 0.8 dB away from the Shannon bound (compared to 2.5 to 3 dB for DVB-S).

Preliminary Remarks

The aforementioned aspects of the new standard influence strongly the way datagrams are dealt with in DVB-S2. In average, longer bearers pack more datagrams together than classical 188-byte MPEG2 containers do, reducing the relative frequency at which segmentation/reassembly of SNDUs should occur. In addition, stronger error protection is expected to decrease dramatically the number of codeword errors at the output of the FEC decoder, and therefore the number of garbled packets upon reassembly as well.

3.4.3 On the BCH Codes of DVB-S2

Hypotheses

Let's consider again the ratio $\eta = P_u/P_d$ between the undetectable and the detectable errors at the output of a BCH decoder, relative to FECFRAMEs (or frames). Given the wide range of frame sizes, a straightforward relation between the frame error rate and the SNDU is harder to precise than for DVB-S, although a $1:10$ ratio seems realistic (that is, one bad frame affects 10 SNDUs in average). As in DVB-S, the essential role of the inner code (LDPC) is to lower the perceived BER at the input of the BCH, for which it will be considered again that the overall FEC error detection capabilities are those of the outer BCH code.

Analytical Considerations

For any chosen FEC rate, an inner LDPC code is concatenated with an outer BCH code, in a scheme integrating again both error correction and detection. The $BCH(n,k)$ codes used in DVB-S2 are all shortened from primitive binary BCH codes with $n = 2^m - 1$, m taking the values 16 and 14 for long frames and short frames, respectively. Finally, $t = 12$ for all the codes applied to short frames, whereas codes used on long frames have $t = 12$, $t = 10$ or $t = 8$, defining 4 big families of BCH codes identified by the couples $(m,t) = (16,12), (16,10), (16,8)$ and $(14,12)$. Kim and Lee [100] have shown that for primitive BCH codes having binomial-like weight distributions, as large subclasses of BCH codes including those used in DVB-S2 do [93], equation (3.4) can be reduced to:

$$P_u(C,\varepsilon) \approx \left[2^{-mt}\sum_{i=0}^{t}\binom{n}{i}\right]\cdot 2^{-nE(\lambda,\varepsilon)} \tag{3.12}$$

where $\lambda n = (t+1)$ and $E(\lambda,\varepsilon)$ is the relative entropy between the binary distribution λ and ε, i.e.

$$E(\lambda,\varepsilon) = \lambda\log_2\left(\frac{\lambda}{\varepsilon}\right) + (1-\lambda)\log_2\left(\frac{1-\lambda}{1-\varepsilon}\right) \tag{3.13}$$

Since P_w is known by equation (3.3) and $P_w = P_u + P_d$, the ratio η can be easily calculated. Unlike for the RS codes of DVB-S, η depends on ε and therefore on E_b/N_0. Its variations using a stand-alone BCH code (without LDPC) for QPSK modulation over an AWGN channel are presented for the 4 families of BCH codes previously introduced in Figure 3.4.

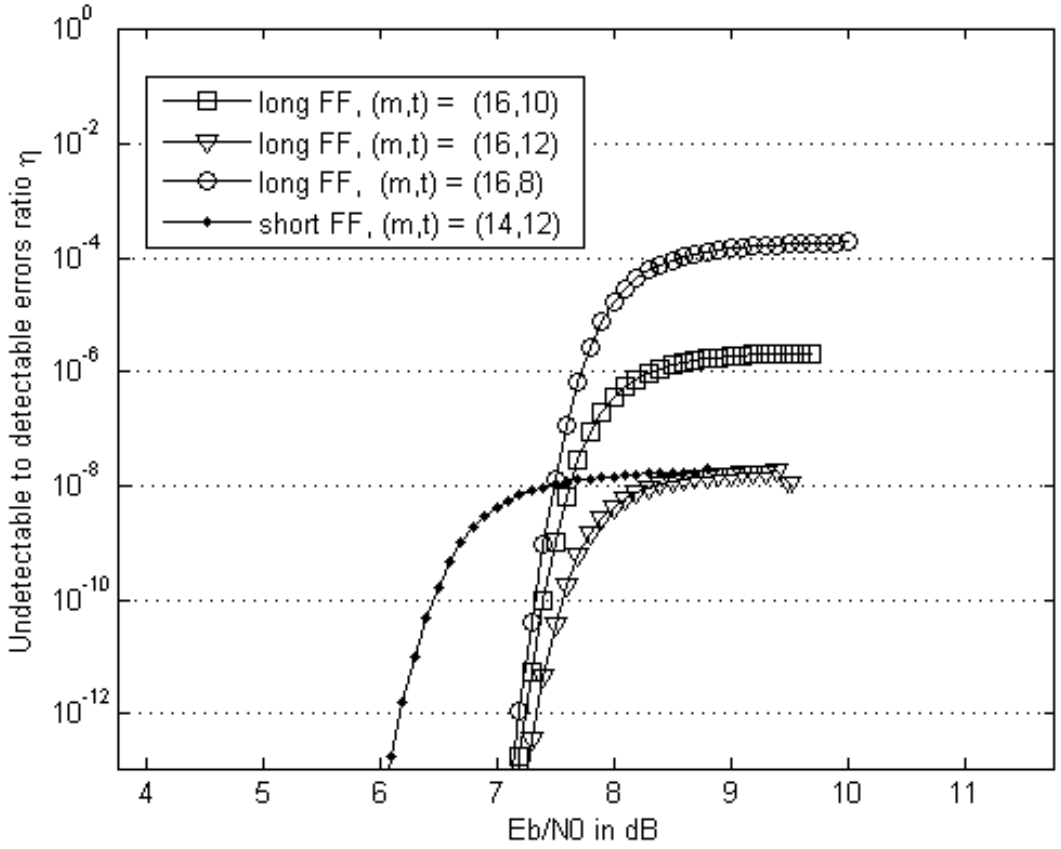


Figure 3.4: Undetectable to detectable errors frequency ratio η for the BCH codes used in DVB-S2 — without the LDPC contribution — over an AWGN channel using QPSK modulation. FF stands for FECFRAME, or frame.

For 17 out of the 21 codes, the ratio between undetectable and detectable errors is lower than 10^{-8} for the whole E_b/N_0 range, reaching its maximum for a given E_b/N_0 value and decreasing rapidly around it. The 4 remaining codes (those with low t) present also low figures for η, between 10^{-4} and 10^{-6}, making their performances similar to those of the Reed-Solomon code in DVB-S. The concatenation with an inner LDPC code is expected to decrease the particular E_b/N_0 value for which the maximum η is reached for every code, without fundamentally changing its variations. Maximum values of η for each code can be found in Table 3.1.

Numerical simulations similar to those done for DVB-S were carried out in order to confirm the above figures. However, due to the very low frequency of the studied phenomena no conclusive results could be derived.

n_{LDPC}	LDPC rate	k_{BCH}	n_{BCH}	m	t	η_{max}
long FF	1/4	16008	16200	16	12	**1.88E-08**
	1/3	21408	21600	16	12	**1.88E-08**
	2/5	25728	25920	16	12	**1.88E-08**
	1/2	32208	32400	16	12	**1.88E-08**
	3/5	38688	38880	16	12	**1.88E-08**
	2/3	43040	43200	16	10	**2.10E-06**
	3/4	48408	48600	16	12	**1.88E-08**
	4/5	51648	51840	16	12	**1.88E-08**
	5/6	53840	54000	16	10	**2.10E-06**
	8/9	57472	57600	16	8	**2.00E-04**
	9/10	58192	58320	16	8	**2.00E-04**
short FF	1/4	3072	3240	14	12	**2.00E-08**
	1/3	5232	5400	14	12	**2.00E-08**
	2/5	6312	6480	14	12	**2.00E-08**
	1/2	7032	7200	14	12	**2.00E-08**
	3/5	9552	9720	14	12	**2.00E-08**
	2/3	10632	10800	14	12	**2.00E-08**
	3/4	11712	11880	14	12	**2.00E-08**
	4/5	12432	12600	14	12	**2.00E-08**
	5/6	13152	13320	14	12	**2.00E-08**
	8/9	14232	14400	14	12	**2.00E-08**
	9/10	na	na	na	na	na

Table 3.1: Maximum values of $\eta = P_u/P_d$ at FECFRAME level for the BCH codes of DVB-S2. The LDPC code rate with which they are concatenated in DVB-S2 is given for informative purposes. FF stands for FECFRAME, or frame.

Post-FEC Bit Error Distribution in DVB-S2 Frames

In parallel to the aforementioned analysis, we also examined closely the behaviour of a DVB-S2 FEC decoder under increasing input error levels.

Our first observation was that for all of the 21 frame types, there was an input error level above which the FEC decoder *suddenly* became unable to keep up with the decoding process, which is coherent with the very steep slopes in the BER vs. E_s/N_0 domain of DVB-S2's FEC (see Figure 4.2). Only a very delicate tuning through a long trial-and-error process of the input error level allowed finding a decoding situation with corrupted and clean frames coexisting, corresponding often to an input BER exceeding 0.2, or 20% — well beyond any realistic functional domain. In other words, there is no "middle point" in which some frames are wrong and some are not in DVB-S2: service losses due to errors are sudden and total for practical matters. Second, we observed that bit errors affecting a corrupted frame are invariably scattered all over it. This is true even at error levels only slightly higher than those causing the decoder to toggle from a decoding to a non-decoding state.

These observations suggest that the corruption of a single frame immediately leads to the loss of its entire payload, regardless of how protected — e.g. by a CRC — its transported SNDUs are. In

addition, it suggests that undertaking the analysis of a corrupted frame in the hope of saving some unaffected SNDUs in its erroneous payload is pointless.

3.4.4 Partial Conclusions and Perspectives

In the case of DVB-S2, the conclusion of this study is twofold.

First, experimental results show that SNDUs invariably share the fate of the frame(s) carrying them. Neither flawed SNDUs inside a somewhat clean frame nor clean SNDUs on corrupted frames exist in practice. For this reason, a *per frame* management of error events seems more suited to DVB-S2 than a *per SNDU* approach, which would be redundant and non optimal. GSE's design choice of not applying a CRC to every single SNDU is therefore justified. Nonetheless, with the new challenges of DVB-S2 come also new concerns and variables to be taken into account as well. The possibility exists e.g. that real-time adaptation of the physical layer to the link conditions may bring new error patterns or unexpected frame corruption/loss that have not been considered here. In order to guarantee unconditional frames validity under such hypotheses, GSE designers have chosen to append CRCs only to those SNDUs being fragmented among two or more link layer frames for caution. In addition, GSE allows an optional CRC_{32} to be calculated *per frame* as described in Section 4.5.

Second, our analyses show that the results obtained for DVB-S can be extended to DVB-S2, allowing GSE to benefit also from the cross-layer enhancements evoked in the DVB-S context. For the 17 codes mentioned above, detectable frame errors are 10^8 times more frequent than undetectable errors, and a bit less for the remaining 4 ones. Since detectable errors are known from the FEC decoder, any CRC in DVB-S2 produces redundant information almost always. For the 17 strongest codes, statistically, defining the QEF target in the same way as for DVB-S ($FER \leq 10^{-7}$ at the input of the demultiplexer), the discarding (or loss) of 10 SNDUs due to an undetected frame error has therefore a probability equal to $10^{-8} \times 10^{-7} = 10^{-15}$, representing an event occurring every 11 000 years of full-time transmission. If the information concerning the nature of the codeword error was taken in account at GSE level before SNDU extraction (e.g. tagging a frame as a detectable FEC error), GSE could then drop it without processing every single SNDU and trigger directly the appropriate decisions — such as e.g. re-asking their missing chunks if ARQ is implemented.

3.5 Conclusion

This chapter assessed the way error control is managed in the lower layers of DVB satellite networks, by studying how FEC and adaptation layer CRCs interact to provide error-free data to the network layer.

Analyses of the error patterns at the output of a DVB-S FEC subsystem at the receiver side showed that the outer Reed-Solomon decoder is aware of the vast majority of frame errors occurring upon decoding and SNDU reassembly, and that resilient or undetectable errors account for less than 10^{-5} (or 0.001%) of the times a CRC check fails in adaptation layers. Unfortunately, this information

is unknown by CRCs, who have to find all the errors on their own after thorough analysis of every single SNDU. This suggests that the bandwidth-consuming task of the SNDU integrity check could be at least partially offloaded to the FEC subsystem, at no extra-cost and safely. This could be done via an intra-host cross-layer mechanism authorizing the FEC decoder to share its decoding information with the adaptation layer, using either an in-band or out-of-band signaling procedure.

On the other hand, GSE's choice not to append a CRC to every single SNDU has been justified under the lights of DVB-S2's enhanced FEC scheme and longer bearers sizes. The application of a CRC per fragmented SNDU under the precautionary principle appears therefore as a sound engineering decision. In addition, it was shown that DVB-S2's enhanced FEC has lowered the ratio of undetectable to detectable errors to 10^{-8} in new generation satellites, making an undetected error event after FEC decoding extremely rare. For this reason, GSE could also benefit from the cross-layer mechanism suggested for DVB-S.

Chapter 4

GSE: A Cross-Layer Friendly Encapsulation for IP over DVB-S2

4.1 Introduction

4.1.1 Foreword

This chapter presents the second and final part of the work done on DVB adaptation layers in this thesis. It specifically describes the motivation and rationale behind the definition of the new Generic Stream Encapsulation (GSE) protocol [25][28] for IP over DVB-S2, and the GSE protocol itself.

The lack of an optimal adaptation layer for IP over DVB-S2 at the beginning of this work motivated the redaction of 3 technical reports [17][18][19]. These early attempts to seize the stakes of an IP over DVB-S2 encapsulation were finally crystallized in the Internet Draft "A Design Rationale for Providing IP Services Over DVB-S2 Links" (*draft-cantillo-ipdvb-s2encaps*) [21] in the first months of 2005. A timely intercession from Thales Alenia Space allowed this document to be taken to the IPDVB Working Group at the 63^{rd} IETF meeting in Paris, which echoed with the parallel activities of other bodies — ESA and DVB in particular [101] — related to what came to be called GSE two years later. In the following months, inputs from ESA, IETF and a wide array of industry and academia researchers helped define the first versions of the encapsulation protocol, building on the set of specifications and requirements described in the aforementioned Internet Draft. So far, work on GSE has progressed well: its definition/standardization has been finalized and its implementation guidelines [102] are on the verge of completion.

Most of the material for this chapter has been taken (sometimes directly) from this Internet Draft, as well as from [23], [25] and [28].

4.1.2 Problem Statement and Chapter Outline

The uses and performances of the Multi Protocol Encapsulation (MPE) [8] and the Unidirectional Lightweight Encapsulation (ULE) [9] have been widely analyzed in the literature, and they are commonly accepted as the standard ways to carry IP datagrams over DVB satellites. Truth is, their design was constrained by the imperatives of using already deployed DVB satellite architectures built over the MPEG2-TS link layer, a technology optimized for media broadcasting and not for IP services delivery. Indeed, MPEG2-TS constraints such as constant bit-rate and constant end-to-end delay are not a must for IP services, which added to the accumulation of multiple overheads undermine IP carriage efficiency.

Recently approved by the European Telecommunications Standards Institute (ETSI), the DVB-S2 architecture uses the most recent advances in physical layer technology, with the unprecedented possibility in DVB networks to carry network layer datagrams without the use of the MPEG2-TS link layer — paving the way to efficient and more flexible IP carriage over satellite links. It appeared soon that the existing mechanisms to encapsulate IP datagrams or Protocol Data Units (PDUs) over DVB-S offered could not fully exploit the innovative features of the new standard, for which a novel encapsulation had to be proposed. The resulting Generic Stream Encapsulation (GSE) has been designed with the specific characteristics of DVB-S2 in mind, providing all the necessary methods to fully exploit its enhanced capacity, reliability and flexibility.

The purpose of this chapter is to expose the rationale behind the original design choices made for GSE under the lights of DVB-S2's new features, explaining GSE's new approach for IP datagrams transmission over DVB satellite links. After a somewhat detailed introduction to DVB-S2, the rationale for the design of the GSE protocol and the protocol itself are presented. Finally, we highlight the way GSE fits into the new standard, stressing the points where it brings originality where previous solutions would fail.

4.2 Overview of DVB-S2

DVB-S2 [2] is the second generation standard for satellite broadcasting, developed by the Digital Video Broadcasting (DVB) Project from 2003 as the successor of the world-wide known DVB-S standard [1] (1993). This architecture is designed for broadband satellite applications such as digital television or radio, as well as interactive services such as Internet access or content distribution.

This section presents an overview of DVB-S2 and its main features. Ampler and more precise information on DVB-S2 can be found in normative References [2] and [103], as well as in the very complete *DVB-S2 Special Issue* of the *International Journal of Satellite Communications and Networking* of April 2004 [80][81][82][104][105][106][107].

4.2.1 DVB-S2 Enhancements over DVB-S

Compared to its predecessor, DVB-S2 features different enhancements both in its physical and link layers.

Physical Layer Enhancements

DVB-S2 implements the most recent developments in modulation and channel coding, with the use of QPSK, 8-PSK, 16-APSK, 32-APSK and especially, the use of concatenated Bose-Chaudhuri-Hocquenghem (BCH) and Low Density Parity Check (LDPC) codes. Although the latter were discovered in 1962 by Gallager [108], their real potential was only re-discovered recently by MacKay and Neal [109][110]. The LDPC code rate can be chosen among 11 values: $^1/_4$, $^1/_3$, $^2/_5$, $^1/_2$, $^3/_5$, $^2/_3$, $^3/_4$, $^4/_5$, $^5/_6$, $^8/_9$ and $^9/_{10}$, for a resulting family of concatenated FEC schemes only 0.4 to 0.8 dB away from the Shannon limit [104], intended to ensure its Quasi Error Free (QEF) target. As for DVB-S, the S2 standard defines QEF as "less than an uncorrected error event per hour", which corresponds to an approximate $FER < 10^{-7}$ after FEC decoding, or an equivalent $BER < 10^{-10}$ [2][82].

Available modulations for DVB-S2 and performance details of its FEC scheme in the PER vs. E_s/N_0 plane are shown in Figure 4.1 and Figure 4.2.

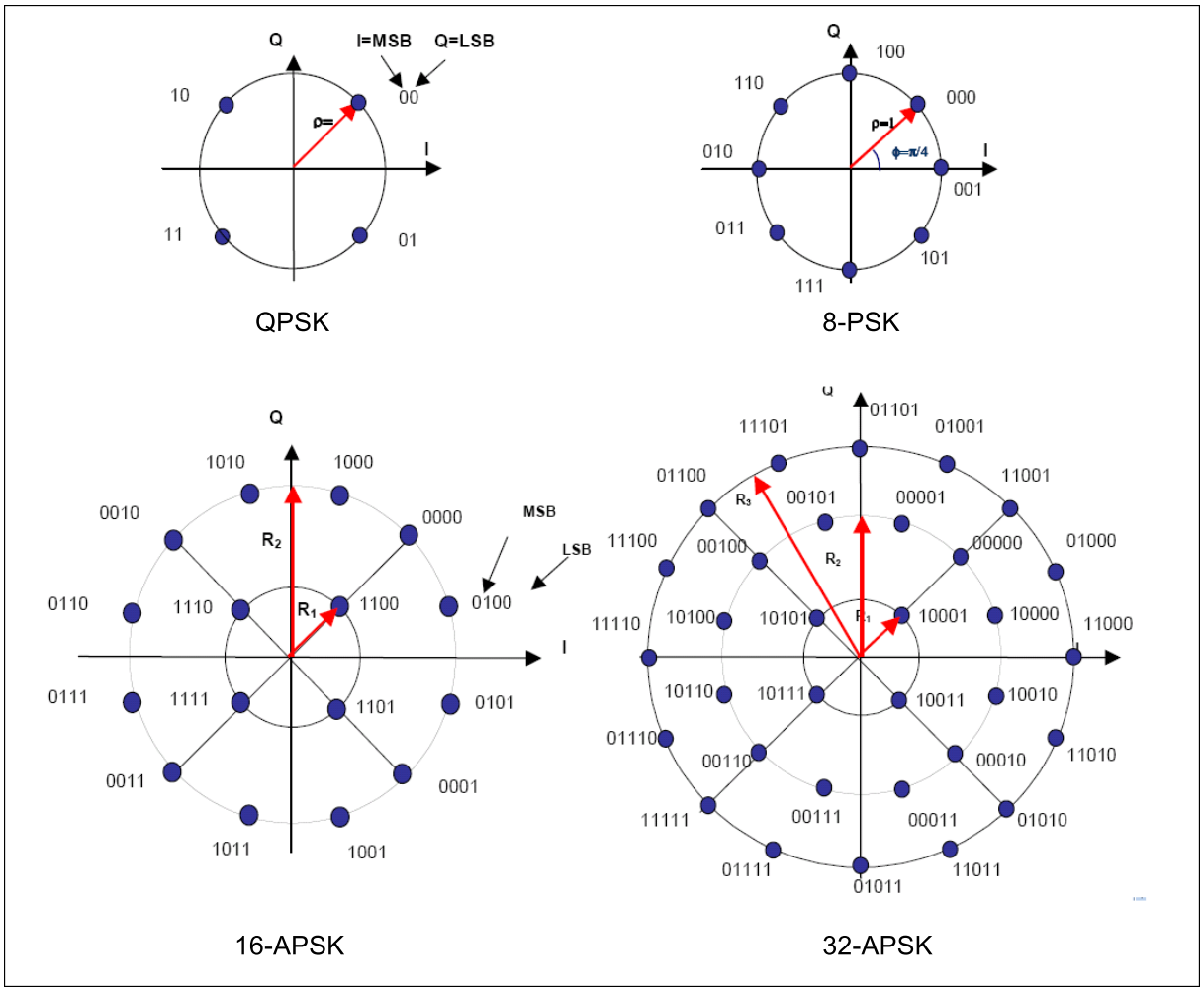


Figure 4.1: The four possible DVB-S2 constellations before physical layer scrambling (source: ETSI).

The combined use of higher order modulations and powerful channel coding allows covering a wide range of E_s/N_0 values from -2.35 dB to 16.05 dB, enlarging considerably the functional domain of

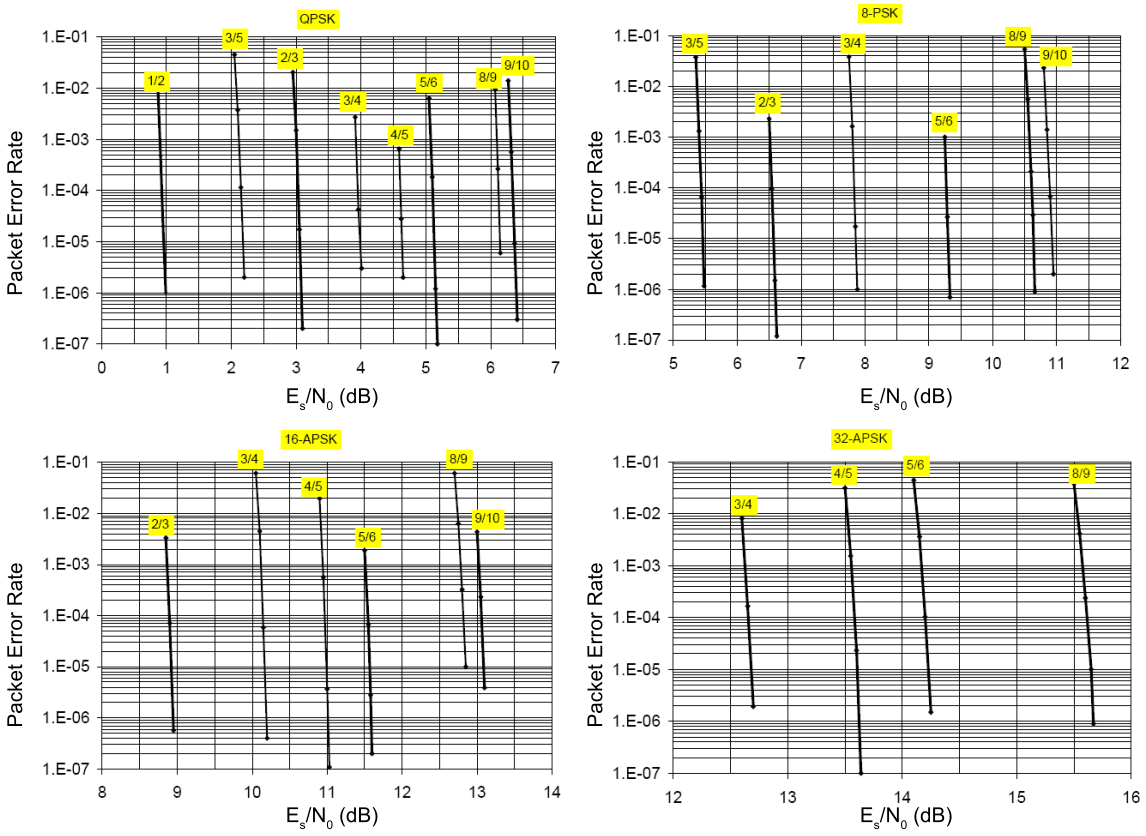


Figure 4.2: Performance of the FEC scheme of DVB-S2 over an AWGN channel, FECFRAME size 64 800 bits (source: ETSI).

the new standard over DVB-S, and increasing *de facto* its raw transmission capacity over more than 40% in terms of spectral efficiency [82][104]. When used for interactive point-to-point applications like IP unicast, theoretical analyses and simulations point out that DVB-S2 performs even better, providing an increase in transmission capacity by a remarkable 150% [80][111].

In order to take full advantage of this flexibility, the new standard provides richer alternatives to the classical Constant Coding and Modulation (CCM) approach. The new Variable Coding and Modulation (VCM) functionality allows 28 different combinations of modulations and error protection levels, labeled as MODCODs to be used and changed on a frame-by-frame basis. This may be combined with the use of a return link — either satellite, such as DVB-RCS [51], or terrestrial — to achieve dynamic closed-loop Adaptive Coding and Modulation (ACM), thus allowing the transmission parameters to be optimized for by a "VCM/ACM manager" for each individual user, on a frame-by-frame basis, according to individual link conditions. This means that the physical layer can provide differentiated QoS levels, a major difference with DVB-S where all receivers shared the same CCM mode.

Note that this allows for QoS requirements from the upper layers (e.g. DiffServ) being mapped into physical layer MODCODs with the help of cross-layer techniques. Although the definition of those mechanisms — including a packet scheduling policy — are out of scope of the design of an encapsulation scheme, an acceptable adaptation layer for DVB-S2 should clearly provide methods to

implement QoS-related scheduling decisions, and to allow for flexible PDU placement and enhanced fragmentation in the flow in order to fully exploit DVB-S2's adaptability. For this reason, since MPE and ULE-like encapsulations provide PDU fragmentation over consecutive bearers (MPEG2 packets) exclusively, their use, although possible, would be suboptimal in the DVB-S2 context.

MODCODs are described in detail in Table 4.1, and their corresponding spectral efficiencies related to Shannon's theoretical limits are represented in Figure 4.3.

MODCOD ID	**Coding and Modulation**		**Spectral Efficiency [bit/s/symbol]**	**Ideal E_s/N_0 [dB] under QEF**
1	QPSK	1/4	0.490	-2.35
2	QPSK	1/3	0.656	-1.24
3	QPSK	2/5	0.789	-0.30
4	QPSK	1/2	0.989	1.00
5	QPSK	3/5	1.188	2.23
6	QPSK	2/3	1.322	3.10
7	QPSK	3/4	1.487	4.03
8	QPSK	4/5	1.587	4.68
9	QPSK	5/6	1.655	5.18
10	QPSK	8/9	1.766	6.20
11	QPSK	9/10	1.789	6.42
12	8PSK	3/5	1.780	5.50
13	8PSK	2/3	1.981	6.62
14	8PSK	3/4	2.228	7.91
15	8PSK	5/6	2.479	9.35
16	8PSK	8/9	2.646	10.69
17	8PSK	9/10	2.679	10.98
18	16APSK	2/3	2.637	8.97
19	16APSK	3/4	2.967	10.21
20	16APSK	4/5	3.166	11.03
21	16APSK	5/6	3.300	11.61
22	16APSK	8/9	3.523	12.89
23	16APSK	9/10	3.567	13.13
24	32APSK	3/4	3.703	12.73
25	32APSK	4/5	3.952	13.64
26	32APSK	5/6	4.120	14.28
27	32APSK	8/9	4.398	15.69
28	32APSK	9/10	4.453	16.05

Table 4.1: MODCOD identifiers and their corresponding spectral efficiencies in information bits/s/symbol under QEF operation. Ideal E_s/N_0 values for each MODCOD are given for indication, assuming code frame size 64800 bits and packet size 188 B. For short coded frames an additional degradation of 0.2 dB to 0.3 dB has to be taken into account (source: ETSI).

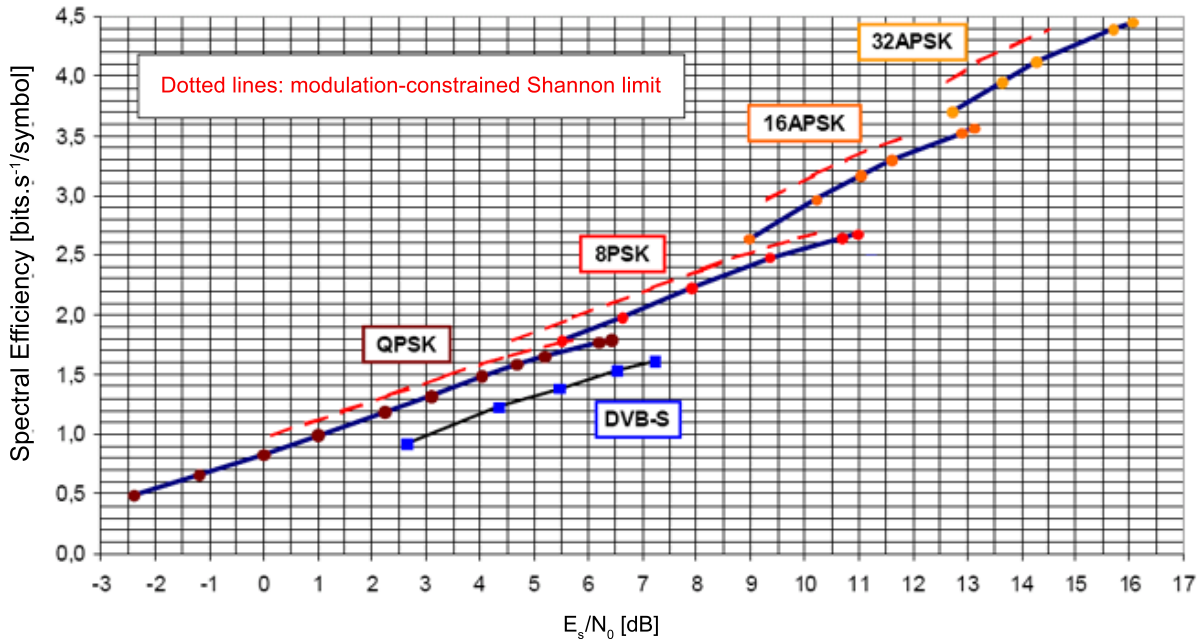


Figure 4.3: Near Shannon limit spectrum efficiency for the DVB-S2 physical layer, obtained by computer simulations on the AWGN channel (ideal demodulator) at Quasi Error Free performance levels $PER = 10^{-7}$ (FECFRAME size 64800 bits, packet size 188 bytes and dummy encapsulation) (source: ETSI).

Link Layer Enhancements

Enhancements of DVB-S2 are not restricted to it physical layer. In addition to the classical packetized MPEG2-Transport Streams [50], DVB-S2 introduces the new *Generic Streams* (GS) above its physical layer, intended to address the non-native way in which network services — such as IP — are carried today over MPEG2-TS using MPE or ULE. Generic Streams can be *packetized* or *continuous*: the former are particularly suited for carrying fixed-length Protocol Data Units (PDU) such as MPEG2 packets or ATM cells, whereas the latter have been designed to accommodate smoothly any kind of input stream format, including continuous bit-streams and PDUs of variable size such as IP datagrams.

As an important addition over DVB-S, before FEC coding both Generic Streams and Transport Streams are tailored into a series of 21 possible *BBFRAMEs* offering different efficiency vs. error protection tradeoffs, with predefined sizes in the range $[384B; 1779B]$ (*short* BBFRAMEs) and $[2001B; 7274B]$ (*long* BBFRAMEs) as shown in figure 4.4. BBFRAMEs' sizes match the input block lengths of the outer BCH codes in DVB-S2, which make them the true basic link-level units of any DVB-S2 stream. Conceptually, in DVB-S2, MPEG2 packets are dealt with as simple PDUs (i.e. as network-level packets) and no longer as link-level bearers, as it was the case in DVB-S.

The choice of continuous Generic Streams for IP datagrams transmissions presents obvious advantages over MPEG2-TS: first, non relevant constraints for interactive services such as constant bit-rate and end-to-end delay can be totally bypassed, allowing for faster and better datagram delivery at reduced overhead and processing complexity. Second, QoS-related rapid changes in the flow structure taken aside, packet fragmentation should occur rather seldom given the large

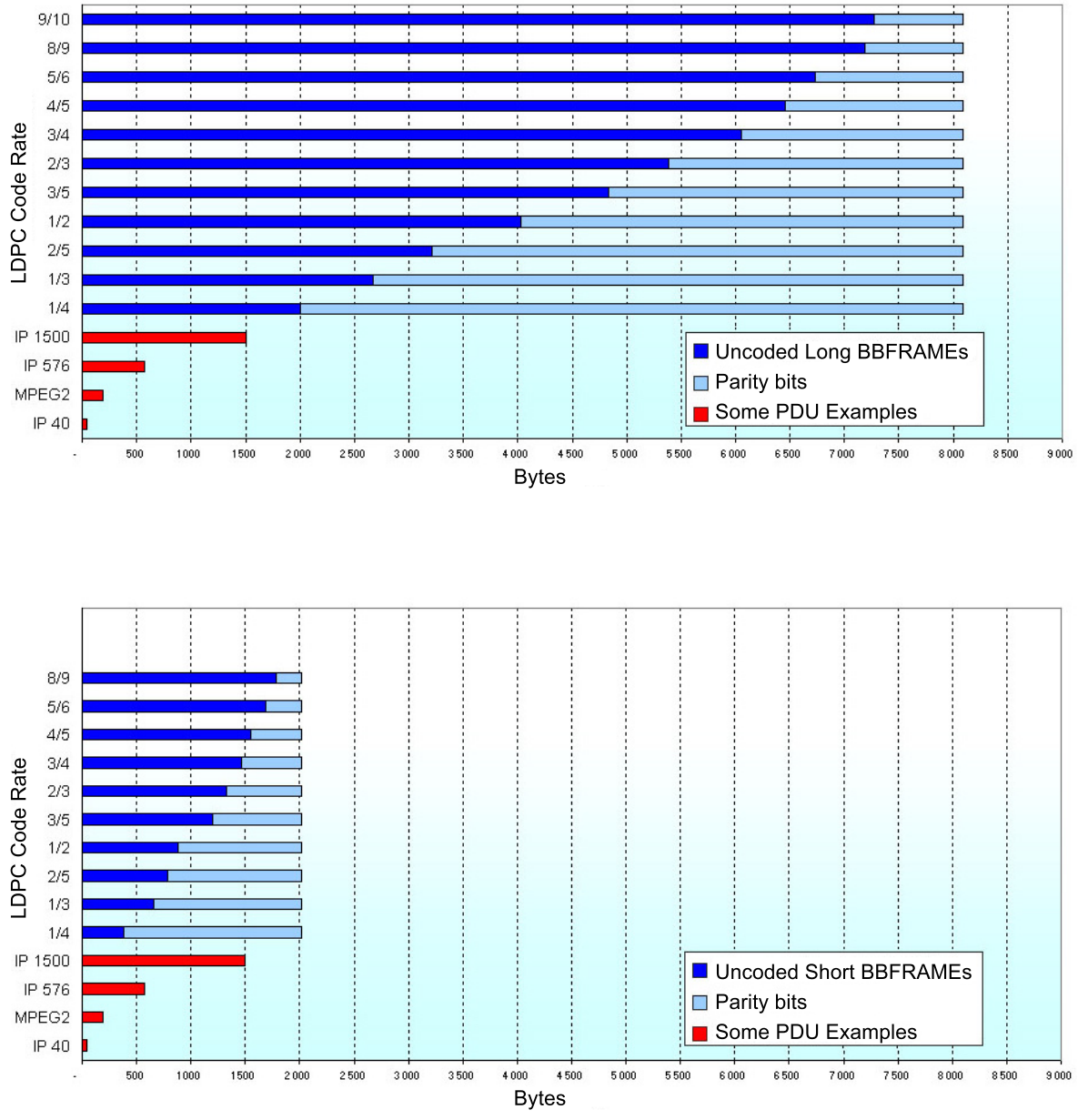


Figure 4.4: Long and short BBFRAMEs in DVB-S2.

BBFRAMEs payload size, up to 40 times broader than a single MPEG2 packet. Measurements on the Internet backbone point out that the frequency-weighted average size of an IP datagram is around 500 bytes [92], so a rough average of $7000 \setminus 500 \simeq 14$ full IP datagrams should in principle be carried in the longest available BBFRAME. In contrast, every single IP datagram suffers in average 2 to 3 fragmentations on top of the MPEG2-TS layer and up to 10 when using ATM. As an obvious and direct consequence, BBFRAMEs using continuous Generic Streams can be expected to accommodate several datagrams in their payloads simultaneously, paving the way for several interesting optimization choices — a quite rare situation today where short link layer payloads are most commonly used.

4.2.2 Functional Blocks in DVB-S2

DVB-S2 is organized as a sequence of functional blocks, summarized in Figure 4.5.

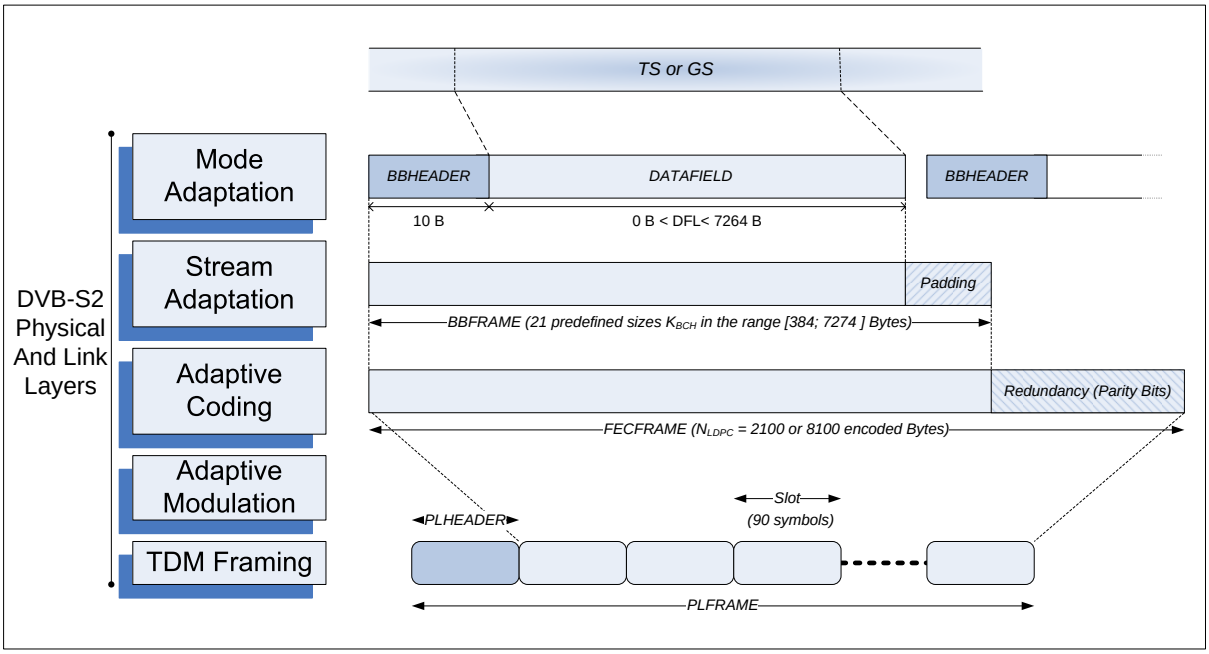


Figure 4.5: Functional blocks of the DVB-S2 standard (source: [103]).

Mode Adaptation

The Mode Adaptation block processes input data structured either as Transport Streams or Generic Streams. Input streams are sliced into DATAFIELDs with size $0 \leq DFL \leq 7264$ bytes to which a 10-Byte BBHEADER is appended. Under VCM or ACM modes, the *maximum length* of every DATAFIELD is chosen dynamically among the 21 possible values in the range [374B;7264B] by the VCM/ACM manager, according to the protection required for each of them. Basically the shorter they are, the more space has been left for FEC redundancy protection. Actual systems may only implement a subset of those 21 sizes.

Stream Adaptation

The Stream Adaptation block is responsible for creating valid BBFRAMEs. For this, it completes every DATAFIELD with Padding when necessary in order to match the length of a valid BBFRAME. BBFRAMEs have one of 21 possible pre-defined sizes in the range [384B;7274B] (in DVB-S, there are only 188 Byte MPEG2 containers at this level). Note that DATAFIELDs sizes are not multiples of 188B: Transport Streams, as well as Generic Streams, are mapped asynchronously over BBFRAMEs.

Adaptive Coding

Adaptive FEC encoding constitutes the third block. A set of coding schemes based on a concatenation of LDPC and BCH codes ensures a very efficient error protection, only 0.4 to 0.8 dB away from the Shannon limit (DVB-S FEC is around 2.5 dB from that margin). In ACM mode, the ACM command dictates dynamically the coding rate to be used for every BBFRAME in order to provide the QEF quality target at the input of the receiver's demultiplexer (see Section 4.2.1). FEC parity bits are calculated and appended systematically to each BBFRAME in order to provide fixed-length FECFRAMEs of 2025B or 8100B (see Figure 4.4). BBFRAMEs with small payloads are completed with more redundancy than those with high payloads, and are therefore more protected.

Adaptive Modulation and Framing

Finally, FECFRAME bits are modulated and raised-cosine filtered, to provide the body of a PLFRAME. 4 different modulations with spectral efficiencies ranging from 2bits/s/Hz to 5 bits/s/Hz are available in DVB-S2. Finally, information about the FEC coding rate and modulation used for every frame (MODCOD) is stored in a PLHEADER and appended to every PLFRAME. Of course, DVB-S2 provides mechanisms to ensure proper reading of every PLHEADER for every receiver without a priori knowledge of the contained MODCOD, so all PLHEADERs use a pre-determined coding and modulation. The final PLFRAME is finally sent over the carrier using classical TDM techniques.

4.2.3 BBHEADER Fields

Several statements in the following sections will refer to fields present in the 10B BBHEADER of every BBFRAME, so a very short description of this entity is presented in Figure 4.6.

The first byte of the MATYPE field specifies whether TS or GS are used, and whether they are packetized, continuous, single or multiple. In the multiple case, the second byte is an "Input Stream Identifier" (**ISI**), whose intended use in the DVB-S2 context for defining logical channels is similar to the one of PIDs for MPEG2.

UPL specifies packet lengths in bits, in the case of packetized input streams. As an example, a value of 0x05E0 (188×8 hexadecimal) is characteristic of MPEG2 packets. According to the

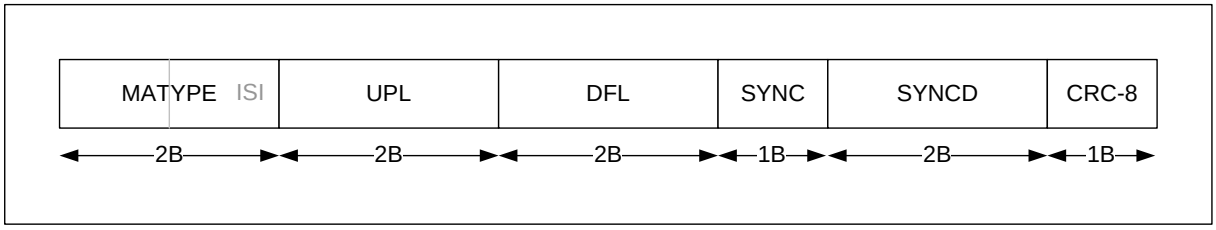


Figure 4.6: A BBHEADER.

standard, a value of 0x0000 indicates a continuous GS. Note however that since the first byte of the MATYPE already indicates the presence of a continuous GS, UPL=0x0000 is delivering a redundant information.

DFL specifies the length of the DATAFIELD actually used in bits, in the range [0b; 58112b] ($58112 = 7264 \times 8$, 7264B being the maximum DATAFIELD length allowed).

SYNC stores the synchronization byte carried by all the packets of a packetized stream, when there is one (e.g. if MPEG2 packets are transported, SYNC=0x47). Since the synchronization byte is carried by BBHEADERs, there is no need for every packet to carry it anymore. A CRC-8 calculated for every packet replaces therefore the synchronization byte in every packet : it is used to validate Segmentation And Reassembly (SAR) applied on them. SYNC is not relevant for continuous Generic Streams.

SYNCD is the distance in bits, for a packetized stream, from the beginning of the DATAFIELD to the first start of packet contained in this DATAFIELD. Its use is therefore similar to a Payload Pointer, as defined in ULE. SYNCD is not relevant for continuous Generic Streams.

Finally, a **CRC-8** is calculated from the previous 9B of the BBHEADER.

Note that BBHEADER fields natively support SAR applied to MPEG2 packets or any other fixed-length packets asynchronously mapped over a BBFRAME flow. Indeed, perfect delineation and reassembly can be achieved by the exclusive use of UPL, DFL and SYNCD for packetized Generic Streams. Finally, the CRC-8 stored in the 1B slot liberated by SYNC in every packet provides an end-to-end integrity check, achieving thus an encapsulation that does not produce any overhead at all (except when Padding is necessary). In DVB-S2, a flow of MPEG2 packets can therefore be sent over a packetized Generic Stream using UPL=0x05E0 and SYNC=0x47.

4.3 Requirements for an Adaptation Layer in DVB-S2

Detailed requirements for transmission of IP datagrams over MPEG2-TS networks have been defined in RFC 4259 [112]. The present section focuses on the requirements for transmission of IP datagrams over DVB-S2 continuous Generic Streams under ACM mode.

The proposed interface should minimize overhead and be simple enough to reduce processing while ensuring adequate network services, as well as be robust to errors and security threats while providing enough flexibility for future extension, as exposed in RFC 3819 [48]. The key goals are

to increase flexibility for IP services and to provide opportunities for better integration of IP-based networks, at reasonable complexity and overhead costs.

4.3.1 Requirements for PDU Encapsulation

Next Level Protocol Type

A key feature of the required encapsulation is to identify the payload type being transported. Such requirement is not specific to DVB-S2: most protocols use a type field to identify a specific process at the next higher layer that is the originator or the recipient of the payload. Given the length of BBFRAMEs, several PDUs will often be packed within the same BBFRAME. Possible ways to differentiate protocol types to which PDUs belong are:

- ISI channels. This requires no overhead and demands that only PDUs from (or to) the same protocol can be sent together in a single BBFRAME. The use of ISI for this purpose can interfere with its use for address resolution or QoS mapping.
- A single Type field per BBFRAME (ex: appended to the BBHEADER or inside it) in an homogeneous traffic environment (e.g. an IPv4-only network). Only homogeneous PDUs (that is, originated or going to a same protocol) will be packed together. This solution produces very small overhead but offers low flexibility for future evolution of the traffic mix.
- A Type field per PDU. In an heterogeneous traffic environment (e.g. a mix of IPv4 and IPv6 packets), it is required that every single PDU is labeled with a proper Type field. This solution produces an overhead proportional to the number of transported PDUs but offers no limits in its flexibility, since the detailed composition of the traffic mix do not affect the encapsulation procedures.

In a context of IPv4 to IPv6 migration and of increased use of the Internet by new applications and users, the last solution seems to be the most adapted. It is also the choice done in ULE.

Integrity Checks

For the IP service, the probability of undetected packet error should be small or negligible, as stated in RFC 3819 [48]. There is therefore a need for a strong error control, either provided by FEC or by other means such as CRCs.

As shown in Chapter 3, the FEC subsystem has been greatly improved in DVB-S2, compared to DVB-S. This single fact makes it worthy to re-evaluate the way integrity checks are usually done in adaptation layers. Under classical MPE or ULE operations, a CRC-32 is appended to each PDU. It is intended to stand as a protection against reassembly errors following corruption or loss of PDU bearers, due to transmission errors or unexpected hardware/software operation. In the DVB-S2 context, the probability of an undetected decoding error has been reduced by several orders of magnitude compared to DVB-S. Frames are usually either correctly decoded, or known to be in

error. This means that an encapsulation protocol for DVB-S2 does not require a CRC for each encapsulated PDU (as in MPE/ULE), saving 4B for each PDU, i.e. around 10% capacity for a classical flow of small packets.

Note that only PDUs with fragments in lost BBFRAMEs will face reassembly problems: a non-fragmented PDU within a lost BBFRAME will be simply lost, even if it had a CRC. In this context it seems adequate to apply CRC integrity checks to the PDUs that may suffer segmentation only.

Link Layer Addressing Capabilities

Individual receivers are not addressable at a BBFRAME level. MPEG2-TS addressing considerations exposed in RFC 4947 [113] apply therefore to BBFRAMEs too and should be used as guidelines for future work on this key topic. These considerations imply the use of an optional Network Point of Attachment (NPA) field appended to every PDU or group of PDUs sharing the same BBFRAME. There are indeed cases where the use of a NPA is required (e.g. when link layer filtering is desired) and if present, it should be carried in a way to allow receivers to pre-filter and discard unwanted PDUs. There are other cases where an NPA is not required (e.g. when a receiver is the end host), and network layer filtering may be used.

An IP over GS interface should therefore support an optional NPA, as ULE does. This field, combined with the logical channel addressing capabilities offered by the ISI field (see Figure 4.6), provides important tools necessary for L2/L3 address resolution issues.

4.3.2 Requirements for Support of Advanced PDU Fragmentation and Packing

Packing Optimization

When ACM or VCM are utilized, successive BBFRAMEs can be sent using different MODCODs. Optimization of system efficiency therefore demands that the transmitter is allowed flexible placement of PDUs in BBFRAMEs (e.g. allowing a partially sent encapsulated PDU to be suspended and resumed in a later, possibly non-consecutive BBFRAME). Since the size and utilization of later BBFRAMEs cannot be predicted, the method must allow an encapsulated PDU to be fragmented more than once. This is a major difference with DVB-S, in which PDU fragments are sent over the next MPEG2 bearer available, regardless of their sizes or required QoS. Furthermore, a receiver should be able to reassemble fragments from several link flows, preferably without requiring a dedicated buffer for each flow (as in ULE/MPE).

Scheduling Issues

MODCOD allocation by the ACM command is closely related to packing optimization, since available DATAFIELD sizes will vary according to the dynamics of the channel. An encapsulation protocol should therefore function smoothly with a scheduling algorithm required to optimize filling and minimize BBFRAME Padding — that may be up to 7264B for an empty DATAFIELD. Such

algorithm should provide ways to fragment, re-order PDUs and delay them when necessary for the sake of optimal filling, but always in the limits of an admissible complexity. In particular, packet re-ordering between different IP flows to optimize BBFRAME filling should be encouraged, while fragment reordering within a single flow of IP packets (that is between 2 fixed ports of 2 end hosts) should be avoided, according to RFC 3819. The scheduling algorithm should take in account the statistical characteristics of the carried IP traffic, and its functioning should not be independent from the ACM command. It should also provide BBFRAME Padding when necessary (when no PDU is ready to be encapsulated).

Precise PDU Delineation and Reassembly

Accurate delineation and identification of scattered PDU fragments must be done by every receiver. As an example, ULE achieves delineation with the joint use of MPEG2's PUSI, a Payload Pointer and a Length field.

Precise PDU delineation is also required for an encapsulation over continuous Generic Streams. The implemented solution may define a ULE-like header for this, but it may also re-use (partially at least) BBHEADER fields that already provide similar functionalities. It should also be robust to synchronization losses, for which an approach using payload pointers and length fields proves desirable. On the other hand, the method must provide ways to ensure reassembly of a scattered PDU even in the case that its fragments are not "adjacent" within 2 consecutive BBFRAMEs, which happens when advanced PDU scheduling/fragmentation procedures are used. In the classical MPEG2-TS/DVB-S scenario, PDU fragmentation is done over MPEG2 packets of the same flow (same multiplex and PID) with Continuity Counters differing only by 1 (modulo 16). This means that a MPE/ULE receiver knows in advance the size and position in the flow of the next PDU chunk needed for proper reassembly. However, in a DVB-S2 context, the scheduling algorithm may chose to send PDU fragments over non-consecutive BBFRAMEs, or place PDU fragments in the middle of a given BBFRAME. Since MPE and ULE do not provide tools to locate scattered PDU fragments with a priori unknown positions and lengths in a BBFRAMEs multiplex, their use in a GS context would clearly be suboptimal.

4.3.3 Requirements for Future Extension

The evolution of the Internet service may in the future require additional functions. A flexible encapsulation protocol should therefore provide a way to introduce new features when required, without having to provide additional out-of-band configuration. A native way to signal header extensions — like the Next-Header protocol type in ULE or the approach used in IPv6 [114] — should be implemented.

4.3.4 Security Requirements

According to RFC 3819, security of the transmitted data must be considered by any link that is intended to support IP. In the DVB-S2 context, security considerations are basically the same for

GS and TS, and are based on those concerning wireless links and MPEG2 networks. IPDVB WG work on ULE security issues is therefore of interest in the DVB-S2 context as well [115][116].

4.3.5 Support for MPEG2 signalling

DVB requirements in [101] state the necessity to allow MPEG2 support within a continuous Generic Stream of heterogeneous PDUs for signaling purposes. Note that DVB-S2 provides a native way for encapsulating MPEG2 packets over *packetized* Generic Streams, but not over *continuous* ones.

4.4 Early Attempts to Meet these Requirements

During the months that followed publication of *draft-cantillo-ipdvb-s2encaps*, an official call for proposals was done by the DVB with modalities defined in [101], and different proposals complying at different levels with the aforementioned requirements were examined. In particular, ESA's "EDGE" [117] and IETF's "GULE" [118] were among the most important ones. It is not completely incorrect to state that, in the end, the best ideas from the different proposals were merged into the resulting GSE protocol.

For informative purposes, we include in Appendix A the guidelines of an early attempt to achieve an efficient and flexible IP encapsulation of IP datagrams over DVB-S2 entirely developed in the context of this PhD work, called EIoSS [19]. It relied on an ambitious all-IP approach making use of extensive cross-layer techniques and achieving quasi overhead-free encapsulation and fragmentation of IP datagrams over continuous Generic Streams. The EIoSS solution was neither published nor commented outside the scope of our close collaborators, and was not intended for submission to DVB's call for proposals. Rather, it was meant to provide a fresh and experimental approach to the problem, in order to stimulate future reflections on innovative IP mappings over any kind of generic stream.

4.5 The Generic Stream Encapsulation Protocol: GSE

[1]Figure 4.7 summarizes GSE operation within DVB-S2's protocol stack, as depicted in [25].

An encapsulated PDU, prefixed by any optional extension ("h") headers added by the Encapsulator, forms the payload of one or more GSE Packets. Each GSE Packet also includes a GSE header ("GH") that contains the length, protocol type and label field (when present). The stream of GSE Packets is placed in the DATAFIELD of a BBFRAME. The sender normally selects the MODCOD (and hence the BBFRAME size) to achieve the QEF target.

The required integrity detection can be achieved using either the error detection capabilities of the FEC coding scheme of the physical layer, or by introducing an optional CRC-32 for each BBFRAME [28]. Each receiver must determine whether the CRC is present. When used, the CRC-32 is

[1]Most of the contents of this section have been taken directly from [25].

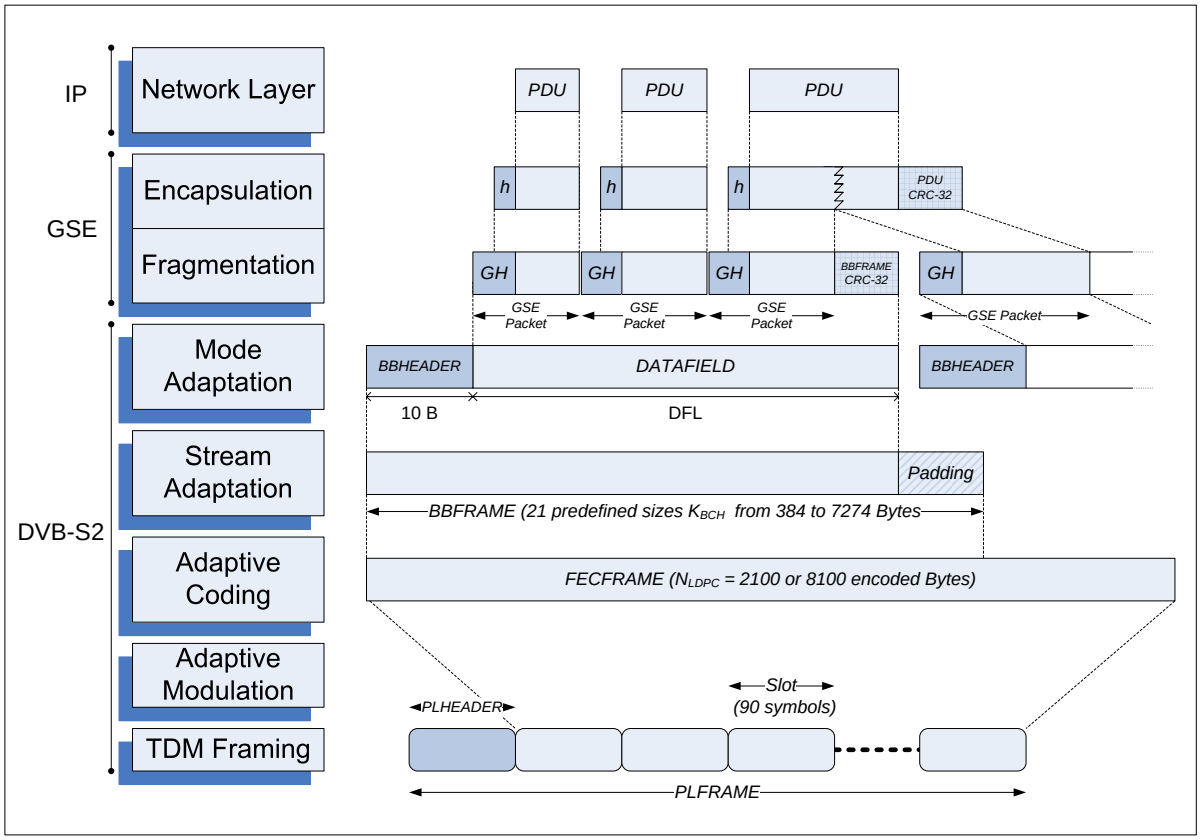


Figure 4.7: Summary of GSE operation within DVB-S2's protocol stack (source: [25]).

placed in the final 4 B of the DATAFIELD (according to the DATAFIELD Length indicated in the BBHEADER). This CRC covers the bytes in the DATAFIELD, incurring an overhead of 0.05% to 1%, depending on the BBFRAME size. This CRC detects residual errors from the FEC decoder with an error probability of $2^{-32} \simeq 10^{-9.6}$, making an undetected event after FEC and CRC under QEF an extremely rare event, ($10^{-7} \times 10^{-9.6} = 10^{-16.6}$ of the time, see Chapter 3).

4.5.1 GSE Encapsulation

A PDU that is sufficiently small may be sent in a single GSE Packet with a 12-bit Length field (indicating the size of the Fragment), a 2-byte Protocol Type field (resembling that in ULE), and an optional address field (Label), preceded by a set of 3 fields in the first 4 bits. When the GSE Packet contains a whole encapsulated PDU, the first two flag bits are set to a value of '1' as shown in Figure 4.8.

A GSE Packet by default carries a 6-byte label (equivalent to the 6-byte address in ULE and MPE). In some cases the label can be suppressed, as in ULE, or may be replaced by a short 3-byte label. Short labels need to be dynamically bound to link layer entities, e.g. using a control protocol. The chosen format is indicated by the flag bits sent in the LT (Label Type) field.

Further GSE Packets may directly follow the first within the DATAFIELD. When the DATAFIELD

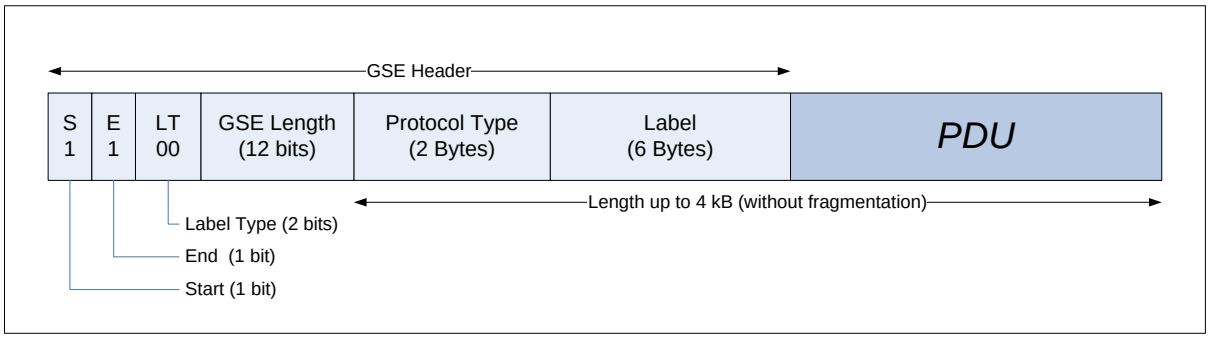


Figure 4.8: Default Generic Stream Packet for a complete encapsulated PDU (S=1, E=1) (source: [25]).

is larger than required to transmit the set of queued GSE Packets, the remaining space (unused) bytes are filled with Padding Bytes (with a value of zero). Padding is silently discarded by the receiver.

In MPEG2 networks, TS Packets can carry network monitoring and control in the form of SI/PSI table sections [50]. Tables are also extensively used in DVB-RCS for a range of functions. The encapsulation therefore provides a method for sending TS Packets [119] within the PDUs sent by a continuous Generic Stream.

4.5.2 GSE Fragmentation and Reassembly

The GSE method permits an encapsulated PDU to be started in a BBFRAME and resumed in a subsequent, possibly non-contiguous, BBFRAME. This fragmentation may be invoked at any time, but is usually used when the encapsulator reaches the end of a BBFRAME and chooses not to pad the remainder of the BBFRAME, but instead to partially send a GSE Packet that will be resumed in another subsequent BBFRAME. This method is also used to send encapsulated PDUs greater than 4 KB, up to the maximum of 64 KB permitted using a 2 byte Total Length field.

To reassemble the fragments, a receiver maintains one reassembly buffer for each encapsulated PDU that may have fragments pending reassembly. Each fragment carries a one-byte fragment identifier, FragID, whose presence is indicated by the setting of the S and E flags. These flags also indicate whether a GSE Packet includes the start, middle, or end of an encapsulated PDU as shown in Figure 4.9. A GSE Packet that contains an S value of 1 carries the start of an encapsulated PDU. The combination S=1, E=0 indicates the presence of a two-byte Total Length field that indicates the size of buffer required to hold the complete encapsulated PDU. This size includes the length of the Label and Type fields and all extension headers that were used.

Each GSE Packet that includes a fragment of a single PDU carries an identical FragID value. When a GSE Packet is received with the combination S=1, E=0, the receiver checks the address (if any) and decides to either accept the fragment and start reassembly or to skip the number of bytes indicated in the Length field. Receivers reassemble the encapsulated PDU, accepting fragments that have a common FragID value, while S=0. A GSE Packet received with (S=0, E=1) indicates that this is the final fragment. The receiver then validates that all parts of a PDU are correctly reassembled by checking both the Total Length of the encapsulated PDU and the CRC-32 in the

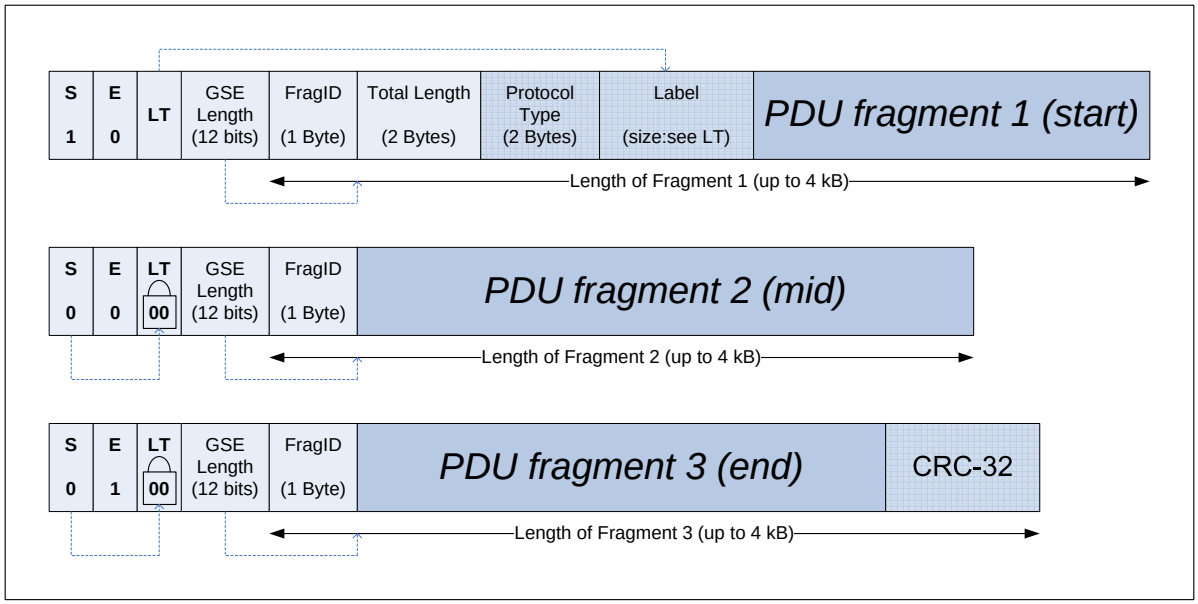


Figure 4.9: Generic Stream Encapsulation for a PDU fragmented into three parts (source: [25]).

GSE Packet	S E	LT	Fields that follow the Length field
Padding	00	00	0 or more bytes of value 00
Resv	00	01	reserved for future use
Mid	00	10	FragID, PDU-part
Resv	00	11	reserved for future use
Resv	01	0x	reserved for future use
End	01	10	FragID, PDU-part, CRC-32
Resv	01	11	reserved for future use
Start	10	0x	FragID, Total Length, Type, Label, PDU-Part
Start	10	10	FragID, Total Length, Type, PDU-Part
Start. LR	10	11	FragID, Total Length, Type, PDU Part
Whole	11	0x	Type, Label, PDU
Whole	11	10	Type, PDU
Whole, LR	11	11	Type, PDU

Table 4.2: Semantics of header flags and corresponding optional fields (source: [25]).

final fragment. Other PDUs addressed to the same or different receivers that belong to different QoS flows may be interleaved between GSE Packets carrying PDU fragments by using a different FragID value. This may be used to minimize the queueing delay of real-time packets.

Receipt of a GSE Packet with S=1 causes any data buffered for the FragID value to be discarded (aborting any incomplete PDU). To bound the waiting time at the receiver and to improve performance after an outage (where a succession of BBFRAMEs failed to be received), fragments that are not completed within 256 consecutive BBFRAMEs are discarded.

GSE Packets can be received with any FragID value. An encapsulation may use a different FragID value for each network flow that it individually schedules. Therefore, for each active ISI value, a receiver must implement a set of 255 reassembly buffers (one for each FragID). The method does not mandate how a FragID value is allocated by a Gateway: only that at any time, each partially sent PDU in the system must use a different FragID value. This could be achieved by associating a specific FragID with each flow, but reuse of a smaller number of FragID values is expected to be more common. An Encapsulator could therefore implement a strategy that constrains the way it schedules GSE Packets and uses FragIDs, to allow the required number of reassembly buffers to be reduced (e.g. the current specification states there must be at least one reassembly buffer for each NPA address in use).

Joint use of the S, E and FragID fields provides a powerful method to assemble scattered PDU pieces upon reception, something neither MPE nor ULE support. Fragments placement in BBFRAMEs can therefore be done with great flexibility, allowing for a great deal of freedom in the scheduler decisions for the sake of system adaptability.

4.5.3 PDU Label Reuse

When several consecutive PDUs are directed to the same receiver(s), redundant information is sent (duplicate addresses) that could be omitted to improve efficiency. This optimization is called Label Reuse and is most beneficial for small PDUs, especially in a trunking scenario, where PDUs are directed to a few receivers.

GSE defines a Label Reuse method that allows a sequence of GSE Packets to be sent to the same receiver without repeating the label. When a receiver re-uses the Label field in a previously received GSE Packet, it receives a GS Packet with the value LT='11'. This labeling mode must not be used for the first GSE Packet sent in a BBFRAME, requiring the label to be sent at least once per BBFRAME. This requirement is primarily relevant for ACM, where a receiver may not have been able to decode the previous BBFRAME(s) (e.g. sent using a less-robust MODCOD, or lost due to propagation impairment) and would therefore not be able to determine whether the label had been modified in these intervening BBFRAMEs. Figure 4.10 illustrates a sequence of PDUs sent with Label Reuse.

Label Reuse reduces the overhead in GSE, with 10 bytes of overhead for the first PDU and only 4 bytes for each subsequent PDUs. This performs a batching of a series of PDUs as a function of the physical layer scheduler, (i.e. the decision about when to concatenate is taken based on the placement of a PDU in a BBFRAME).

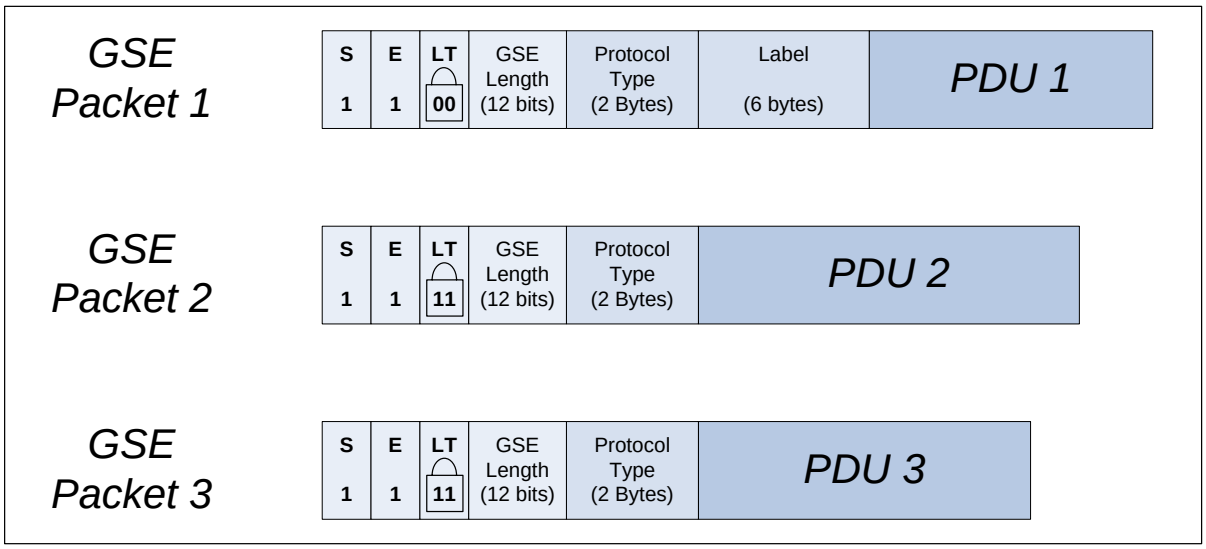


Figure 4.10: Label Reuse for three successive GSE Packets (source: [25]).

4.5.4 GSE Extension Headers

The GSE protocol minimizes the number of fields that need to be processed by a receiver. This design philosophy requires a critical review of functions, removing optional functions to extension headers. This not only simplifies the base design and promotes interoperability between implementations, it also allows the system to be extended after the base protocol is introduced. Separating extensions also permits these functions to be processed (possibly as a part of an upper-layer driver) after framing, integrity checks and NPA address (label) filtering. This method resembles the design of IPv6 [112][114].

In GSE, extension headers are identified by the Type field in the encapsulation header [9][119]. Extension headers can be used to support new types of data and to indicate different processing required for the PDU(s) carried in the encapsulated PDU. ULE defines a base set of extensions [9][119] (e.g. bridging and test formats). These extension headers appropriate to both GSE and ULE.

The TS-Concat extension enables GSE Packets to carry MPEG2 service information, allowing control data to pass over the GS. This uses a Type field of 2 to (TS-Concat Type) to identify a PDU that carries a batch of MPEG2 TS Packets to a common (e.g. multicast) NPA address. The Length field is used by the receiver to determine the number of encapsulated TS Packets as shown in Figure 4.11.

A generalization of the above format allows concatenation of other types of PDU at the encapsulation layer [119]. This uses a Type field with a value of 3, assigned to the PDU-Concat Type field. As in the case for MPEG2 TS Packets, all concatenated PDUs must be sent with the same Label, and must have the same Type (e.g. all be IPv6 packets). In addition, since these concatenated PDUs are not necessarily of the same size, each PDU is prefixed by a two-byte PDU-Length field.

Although Figure 4.11 shows a concatenated payload sent in a single GSE Packet, this is not a

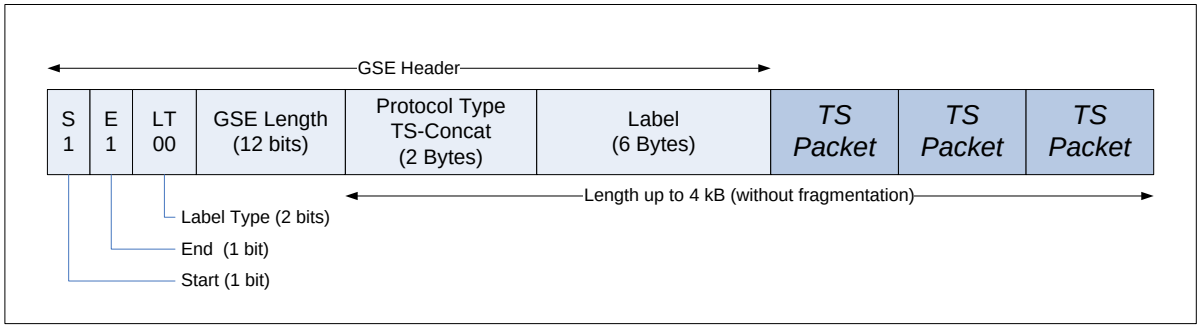


Figure 4.11: Concatenation of TS Packets using the Type field (source: [25]).

requirement, and fragmentation may be used, as in other types of PDU. This allows the encapsulation process to decide whether to concatenate PDUs, and postpones the decision of whether to use one or more GSE Packets to the Scheduler that constructs a BBFRAME DATAFIELD. At the receiver, the cost of processing a concatenated PDU is reduced, in that a receiver accepts all or none of the batch of concatenated PDUs after reading the label in a single GSE Header. This format is also defined for ULE.

4.5.5 On Overhead in GSE

Given the large sizes of BBFRAMEs (up to 7 kB), overhead has become a less important concern in GSE than for ULE or MPE, where link layer frames were about the same sizes of the transported PDUs. Design concerns in GSE have focused on fragmentation flexibility and exploitation of the adaptive features of GSE, rather than on the minimization of overhead and other unused bits. This explains apparent inefficiencies from an overhead point of view, such as the encapsulation of PDU fragments.

GSE designers have however made considerable steps towards the reduction of header overhead (which can be minimized to 2 bytes in particular contexts) with the elimination of CRCs for non-fragmented PDUs. At its early definition stages, GSE designers agreed on a series of flexibility levels in order to limit header overhead. In particular, the final set of supported fragmentations schemes was the center of intense debates, since this particular issue had a major impact on the targeted flexibility — and thus its required price in terms of overhead.

Analytically and experimentally, studying GSE efficiency requires making precise assumptions on link conditions, traffic characteristics, BBFRAMEs sizes, scheduling policies and ACM behaviour. This is a huge difference with PDUs mapped over a series of organized MPEG2 containers under CCM in DVB-S, where easy and straightforward comparisons between ULE and MPE could be done. Under precise system configurations and traffic assumptions, [25], [120] and [121] situate overall overhead values — calculated as a ratio between useful (non-header, non-padding, non-CRC) and sent bits — for GSE in the orders of 1% to 5% (resp. efficiency between 95% and 99%). Undoubtedly, compared to MPE and even to ULE, these are good values.

However, in the author's opinion, such comparisons are to be taken carefully, at least for two reasons. First, the underlying hypotheses regarding the scheduling policy, the traffic size distribution,

the options chosen (e.g. PDU-Concat, Label Reuse) and the series of MODCODs used have a huge impact on the overall overhead figures, which makes fair comparisons under different environments delicate. In contrast, MPE and ULE were easy to compare between them due to the absence of fancy encapsulation options and scheduling considerations, the use of CCM and the inherent organized nature of the MPEG2-TS. Second, large absolute overhead values that would have been considered high under MPE and ULE seem small in the DVB-S2 context, thanks to the large BBFRAMEs sizes available. Take for instance the fragmentation shown in Figure 4.9 and assume the Label field contains a 6-bytes long MAC address. The total required overhead (header information plus CRC-32) for this PDU is 23 bytes, which is almost 3 times the length of a classical ULE header. Using 184-byte MPEG2 payloads the efficiency would have been around 87.5%, whereas using the longest available BBFRAME it rises up to 99.7%.

4.6 Future Developments for GSE

Undoubtedly, the evolution of wireless and satellite networks will allow or require richer functionalities to be added to GSE in the years to come. The native mechanism that could be used for this purpose is described in Section 4.5.4. On top of the existing extension headers already defined, other potential uses of this mechanism include support for compression, QoS-signalling and performance monitoring. For instance, an extension to the encapsulation is being considered to provide confidentiality (encryption) and optional source authentication [116].

4.6.1 GSE Adaptation to other DVB Radio Layers

In particular, additional functions may be provided in the near future to adapt GSE to other DVB radio layers, either existing e.g. DVB-SH [41][42] or to come, such as evolutions of the DVB-H [40], DVB-RCS [51] or DVB-T [122] standards.

4.6.2 BBHEADER Bits Re-Use

Note that among the 10 bytes of BBFRAME headers, at least three (SYNC and SYNCD) are not relevant for continuous Generic Streams, and two (UPL) are redundant (see Section 4.2.3). Indeed, their use has been defined in the DVB-S2 standard for the sole purpose of allowing native transport of fixed-length PDUs over *packetized* Generic Streams. Their re-definition and use in the context of *continuous* Generic Streams might prove useful, and pave the way for further optimizations of future versions of the GSE protocol. Possible uses include: allowing further flow organization, stamping BBFRAMEs for e.g. Operation and Management (OAM) purposes or adapting MODCOD selection based on network layer QoS signaling.

4.6.3 Cross-Layer Enhancement of GSE's Error Control Techniques

It was shown in Chapter 3 that DVB-S2's enhanced FEC has lowered the ratio of undetectable to detectable errors to 10^{-8} in new generation satellites, making an undetected error event after FEC decoding extremely rare. For this reason, GSE could also benefit from the cross-layer mechanisms suggested here for DVB-S.

4.7 Conclusions

This chapter presented GSE, the new encapsulation protocol for IP over DVB-S2 that makes efficient and full use of its enhanced and innovative cross-layer features. The new standard defines a set of advanced coding and modulation waveforms that offer a significant improvement over that provided by DVB-S, a new link layer based on the continuous Generic Stream and especially support for ACM.

GSE allows a transmitter to directly transport network packets in BBFRAMEs. The design rationale has been presented and the header format explained. Although GSE improves system performance by reducing the encapsulation overhead compared to the one required for MPEG2-TS over DVB-S, its most significant performance benefit arises from the flexible placement and fragmentation of packets particularly when using ACM. This flexibility allows operators to change the waveform on a frame-by-frame basis, providing an important reduction in the cost of providing the service. In this sense, GSE is truly a cross-layer friendly encapsulation protocol.

GSE has been standardized by the Digital Video Broadcast (DVB) Technical Module, and its implementation guidelines are on the verge of completion. It is well-suited as an IP-friendly encapsulation for DVB-S2, and active work aims to extending its scope to other DVB radio layers.

Chapter 5

HERACLES: Header Redundancy Assisted Cross-Layered Error Suppression

"Any redundancy in the source will usually help if it is utilized at the receiving point. In particular, if the source already has a certain redundancy and no attempt is made to eliminate it in matching to the channel, this redundancy will help combat noise."

Claude Shannon, 1948

5.1 Introduction

5.1.1 Redundancy, Compression and Robustness

We all are more or less familiar with the notion of *redundancy*: it commonly relates to the degree to which elements of a given system are superfluous or unnecessary. Since redundant elements classically consume costly resources without contributing to the overall system, maximizing an economic value function related to the exploitation of the system generally benefits from redundancy reduction or *compression*. For the engineer, however, redundancy bears more than a compression potential: redundant systems are known to be more robust and reliable in the presence of aleatory failures, for which they are widely used e.g. in critical life-support or flight control systems, by intentionally degrading a resource budget.

Digital communications systems lie somewhere in between these two realities, since they need to transmit messages reliably, while making the most efficient use of scarce resources such as power and bandwidth. Shannon's famous separation theorem [7] states that source compression and channel coding should be separated, and that error free transmission is possible as long as the entropy of the source is less than the capacity of the channel. Source coding reduces the natural redundancy of the message, whereas channel coding introduces artificial, wanted redundancy in order to cope with channel errors. Although this is optimal in the information theoretical sense

with very long source and channel codes among other conditions, real communications on very bad channels require much added FEC redundancy — and sometimes, retransmissions — which at the end raise the question of the net gain for the overall system. This can be particularly true in the case of highly redundant and short messages, such as those produced by many current protocol stacks. Recent works have therefore started analyzing new approaches to the general transmission problem and hence proposing new alternatives for redundancy management. Among them, the emerging current of the joint source-channel coding [13][14] and the framework of the Discrete Universal Denoiser (DUDE) [123] seem the most promising ones.

Following a similar reasoning, this chapter aims at exploring some original possibilities offered by the natural redundancy of messages produced by common protocol stacks others than compression. It introduces HERACLES (Header Redundancy Assisted Cross-Layered Error Suppression), a novel intra-host cross-layer mechanism making use of such redundancy at the *lower layers* of the protocol stack, that achieves enhanced link-layer framing/delineation performances and increased overall error protection without added overhead.

5.1.2 Header Redundancy in Common Protocol Stacks

On Header Compression

Header compression schemes such as Van Jacobson's CTCP [124], Degermark's IPHC [125][126], CRTP [127] and ROHC [128] build their bandwidth saving abilities on the analysis of header contents and variability. The main reason why header compression can be done at all is the fact that there is significant redundancy between header fields, not only within the header itself but in particular between consecutive packets belonging to the same *packet stream*. By sending static field information only initially and utilizing dependencies and predictability for other fields, the header size can be significantly reduced for most packets. As a first approximation, a packet stream could be defined as the set of packets sent from a particular address and port to a particular destination address and port using the same protocol stack. The need for replicated information in packets belonging to the same stream is a direct consequence of the original design choices done for IP, which requires independent datagram processing and routing at every network node.

In the early years of IP deployment, speed, scalability and bandwidth savings were not primary issues, since the new memoryless and connectionless protocol was deployed in sparse host environments over wired links, mainly for research purposes. Over the years, the increasing the popularity of IP services and following the massive development of wireless and in resource-constrained IP-based links, bandwidth competition became a primary concern for network managers and operators. It soon appeared that the accumulation of structural overhead — which reached up to 90% for some applications — strongly undermined efficiency, thus impairing efficient network usage and plummeting the average revenue per user. This naturally led to the first works aimed to reduce header redundancy through compression, resulting in the above-mentioned schemes.

As a concluding remark, note that two major trends threaten to decrease the efficiency of IP over wireless links, making works on header redundancy even more topical. The first one is related to mobility : current schemes allowing a host to keep its original IP address when it has moved in the network usually require tunneling-type encapsulations, doubling at least header sizes. The second

one is the coming of IPv6, clearly intended for high-capacity networks where the expanded header size becomes negligible due to the large amounts of data transmitted.

Variability of Header Fields

Just like header compression schemes do, the cross-layer mechanism we introduce in this chapter builds its abilities on the analysis of header contents and variability, although its purpose is not related to compression. For this reason, a general insight to classical headers structure — useful for both frameworks — constitutes an excellent starting point for the developments of this chapter.

Generally speaking, header bytes of any protocol stack can be classified into three main categories[1].

STATIC: Fields that are expected to be constant throughout the lifetime of the packet stream. Some of them define the protocol stack, and some others are specific to a given connection. Examples are source and destination addresses, ports, channel information and protocol type fields.

INFERRED: Information contained in these header fields can be deduced directly or indirectly from other values, such as checksums or packet sizes.

CHANGING: For a given receiver, those fields appear to vary randomly. Those include TTL, type of service, sequence numbers, timestamps etc.

For most protocol stacks, STATIC fields account for the majority of bytes in header fields. For instance, take classical IP/UDP/RTP or IP/TCP transmissions, for which tables 5.1 and 5.2 present real values taken from RFC 3095 [129] and RFC 4413 [130].

	IPv6/UDP/RTP	**IPv4/UDP/RTP**
STATIC	44.5 **(74%)**	20.5 **(51%)**
INFERRED	4 (7%)	6 (15%)
CHANGING	11.5 (19%)	13.5 (34%)
TOTAL	60 (100%)	40 (100%)

Table 5.1: Summary of field categories for IP/UDP/RTP (source: RFC 3095).

	IPv6/TCP	**IPv4/TCP**
STATIC	40 **(67%)**	15.75 **(39%)**
INFERRED	2.5 (4%)	4.5 (11%)
CHANGING	17.5 (29%)	19.75 (50%)
TOTAL	60 (100%)	40 (100%)

Table 5.2: Summary of field categories for IP/TCP (source: RFC 4413).

Similar studies can be done for every protocol stack, and in particular for ATM-based ones.

[1] Simplified version from ROHC's classification [129], which uses 5 categories: STATIC, STATIC-DEF, STATIC-KNOWN, INFERRED AND CHANGING. It differs only slightly from the one introduced by Degermark in [126].

Application Example: FTP Download with Ethernet/IPv4/TCP

In order to illustrate the previously presented classification, Figure 5.1 presents some successive frames captured at the Ethernet driver of my computer acting as a client, showing the first stages of a FTP control connection with a remote server.

In this precise example, the overall 54-byte long header is constituted from the Ethernet, the IPv4 and the TCP headers. Figure 5.2 presents a zoom of the first frame in Figure 5.1, detailing the structure of the combined Ethernet/IPv4/TCP header.

From RFC 3095 and RFC 4413, details for each of these fields are presented as follows :

Eth1-Eth2 : Ethernet Destination/Source Addresses (12 bytes). These fields are part of the definition of a stream and must thus be constant for all packets in the stream. The fields are therefore classified as STATIC.

Eth3 : Type (2 bytes). This field will usually have the same value in all packets of a packet stream. It encodes the type of the subsequent header. The field is therefore classified as STATIC.

IP1 : IP Version and Header Length (1 byte). The version field (4 bits) states which IP version is used. Packets with different values in this field must be handled by different IP stacks, and therefore all packets of a packet stream must be of the same IP version. Concerning the header length field (4 bits), as long as no options are present in the IP header, the header size is constant and well known (20 bytes). Accordingly, the combined Version/Header Length byte is STATIC.

IP2 : IP Type of Service (1 byte). This field might be expected to vary during the lifetime of a packet stream. CHANGING.

IP3 : Packet Length (2 bytes). Information about packet length is expected to be provided by the link layer. The field is therefore classified as INFERRED.

IP4 : IP Identification (2 bytes). The IPv4 specification does not describe exactly how this field is to be assigned values, but only that each packet should get an IP ID that is unique for the source-destination pair and protocol for the time during which the datagram (or any of its fragments) could be alive in the network. Assignment of this value can be done in several ways, for which this field is globally CHANGING.

IP5 : Flags and Fragment Offset (2 bytes). According to RFC 4413, the Don't Fragment flag and the Reserved flag are CHANGING (differing from RFC 3095, which considers the former STATIC), but the More Fragments flags is STATIC. As for the 13 bits of the Fragment Offset, under the ideal assumption that no fragmentation occurs, they are always zero. If fragmentation were to be further considered, only the first fragment would contain the TCP header, and the fragment offset of this packet would still be zero. Summarizing, 14 out of 16 bits in IP5 are STATIC.

IP6 : Time To Live (1 byte). In general this field is expected to be constant during the lifetime of a packet stream. However it can alternate between a limited number of values due to route changes, for which it is classified as CHANGING.

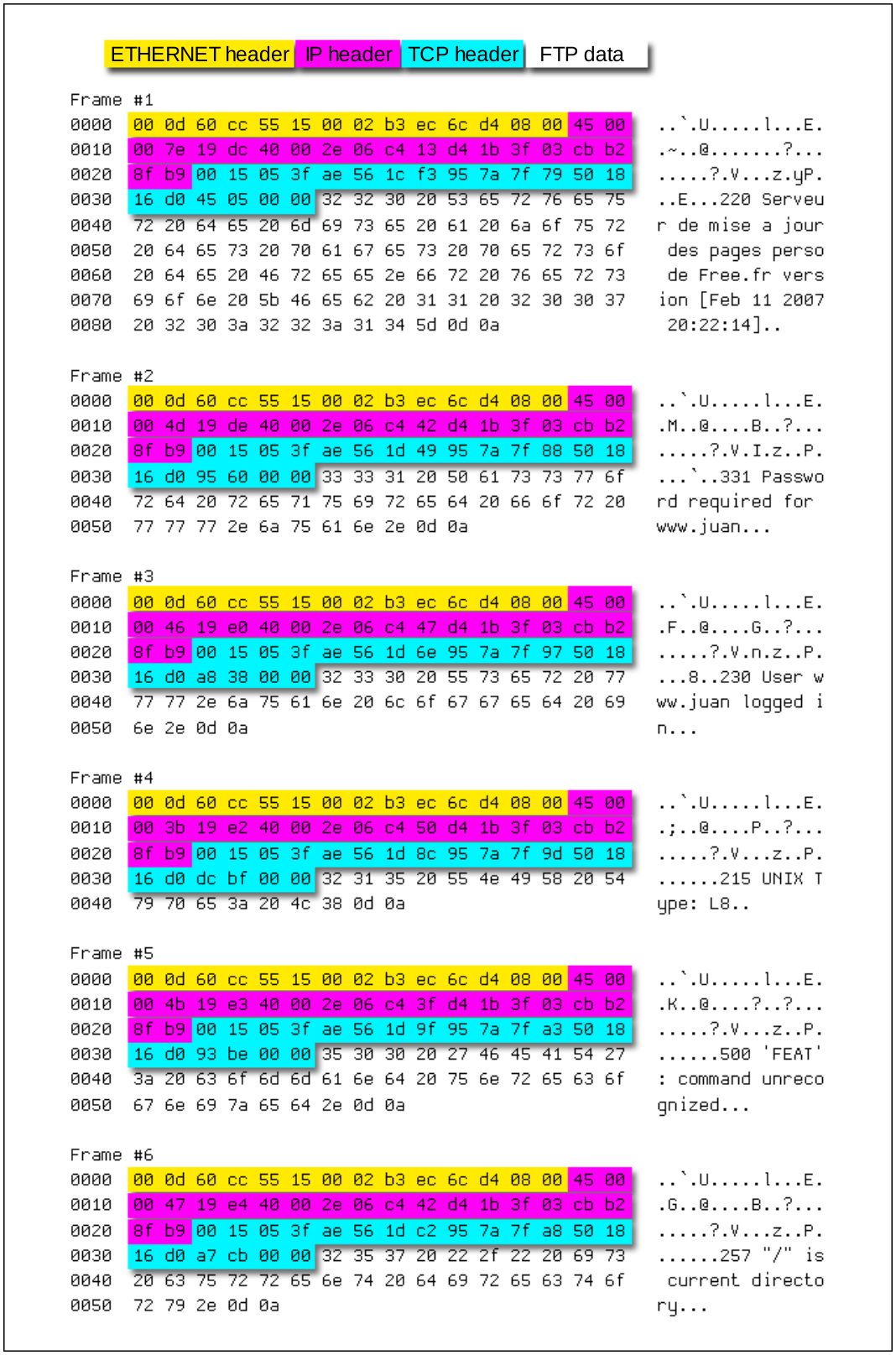
IP7 : Protocol (1 byte). This field will usually have the same value in all packets of a packet stream. It encodes the type of the subsequent header. Only where the sequence of headers changes (e.g., an extension header is inserted or deleted or a tunnel header is added or removed) will the field change its value. The field is therefore classified as STATIC.

IP8 : Header Checksum (2 bytes). This field is directly calculated from selected header bits, for which it is classified as INFERRED.

IP9-IP10 : Source/Destination Addresses (8 bytes). These fields are part of the definition of a stream and must thus be constant for all packets in the stream. The fields are therefore classified as STATIC.

TCP1-TCP2 : Source/Destination Ports (4 bytes). These fields are part of the definition of a stream and must thus be constant for all packets in the stream. The fields are therefore classified as STATIC.

TCP3-TCP4 : Sequence/Acknowledgement Number (8 bytes). These fields are incremented during the progress of the transmission, for which they clearly are CHANGING.




```
Frame #1
0000  00 0d 60 cc 55 15 00 02 b3 ec 6c d4 08 00 45 00   ..`.U.....l...E.
0010  00 7e 19 dc 40 00 2e 06 c4 13 d4 1b 3f 03 cb b2   .~..@.......?...
0020  8f b9 00 15 05 3f ae 56 1c f3 95 7a 7f 79 50 18   .....?.V...z.yP.
0030  16 d0 45 05 00 00 32 32 30 20 53 65 72 76 65 75   ..E...220 Serveu
0040  72 20 64 65 20 6d 69 73 65 20 61 20 6a 6f 75 72   r de mise a jour
0050  20 64 65 73 20 70 61 67 65 73 20 70 65 72 73 6f    des pages perso
0060  20 64 65 20 46 72 65 65 2e 66 72 20 76 65 72 73    de Free.fr vers
0070  69 6f 6e 20 5b 46 65 62 20 31 31 20 32 30 30 37   ion [Feb 11 2007
0080  20 32 30 3a 32 32 3a 31 34 5d 0d 0a                20:22:14]..

Frame #2
0000  00 0d 60 cc 55 15 00 02 b3 ec 6c d4 08 00 45 00   ..`.U.....l...E.
0010  00 4d 19 de 40 00 2e 06 c4 42 d4 1b 3f 03 cb b2   .M..@....B..?...
0020  8f b9 00 15 05 3f ae 56 1d 49 95 7a 7f 88 50 18   .....?.V.I.z..P.
0030  16 d0 95 60 00 00 33 33 31 20 50 61 73 73 77 6f   ...`..331 Passwo
0040  72 64 20 72 65 71 75 69 72 65 64 20 66 6f 72 20   rd required for
0050  77 77 77 2e 6a 75 61 6e 2e 0d 0a                  www.juan...

Frame #3
0000  00 0d 60 cc 55 15 00 02 b3 ec 6c d4 08 00 45 00   ..`.U.....l...E.
0010  00 46 19 e0 40 00 2e 06 c4 47 d4 1b 3f 03 cb b2   .F..@....G..?...
0020  8f b9 00 15 05 3f ae 56 1d 6e 95 7a 7f 97 50 18   .....?.V.n.z..P.
0030  16 d0 a8 38 00 00 32 33 30 20 55 73 65 72 20 77   ...8..230 User w
0040  77 77 2e 6a 75 61 6e 20 6c 6f 67 67 65 64 20 69   ww.juan logged i
0050  6e 2e 0d 0a                                       n...

Frame #4
0000  00 0d 60 cc 55 15 00 02 b3 ec 6c d4 08 00 45 00   ..`.U.....l...E.
0010  00 3b 19 e2 40 00 2e 06 c4 50 d4 1b 3f 03 cb b2   .;..@....P..?...
0020  8f b9 00 15 05 3f ae 56 1d 8c 95 7a 7f 9d 50 18   .....?.V...z..P.
0030  16 d0 dc bf 00 00 32 31 35 20 55 4e 49 58 20 54   ......215 UNIX T
0040  79 70 65 3a 20 4c 38 0d 0a                        ype: L8..

Frame #5
0000  00 0d 60 cc 55 15 00 02 b3 ec 6c d4 08 00 45 00   ..`.U.....l...E.
0010  00 4b 19 e3 40 00 2e 06 c4 3f d4 1b 3f 03 cb b2   .K..@....?..?...
0020  8f b9 00 15 05 3f ae 56 1d 9f 95 7a 7f a3 50 18   .....?.V...z..P.
0030  16 d0 93 be 00 00 35 30 30 20 27 46 45 41 54 27   ......500 'FEAT'
0040  3a 20 63 6f 6d 6d 61 6e 64 20 75 6e 72 65 63 6f   : command unreco
0050  67 6e 69 7a 65 64 2e 0d 0a                        gnized...

Frame #6
0000  00 0d 60 cc 55 15 00 02 b3 ec 6c d4 08 00 45 00   ..`.U.....l...E.
0010  00 47 19 e4 40 00 2e 06 c4 42 d4 1b 3f 03 cb b2   .G..@....B..?...
0020  8f b9 00 15 05 3f ae 56 1d c2 95 7a 7f a8 50 18   .....?.V...z..P.
0030  16 d0 a7 cb 00 00 32 35 37 20 22 2f 22 20 69 73   ......257 "/" is
0040  20 63 75 72 72 65 6e 74 20 64 69 72 65 63 74 6f    current directo
0050  72 79 2e 0d 0a                                    ry...
```

Figure 5.1: Hexadecimal dump at Ethernet level of incoming packets in a FTP download.

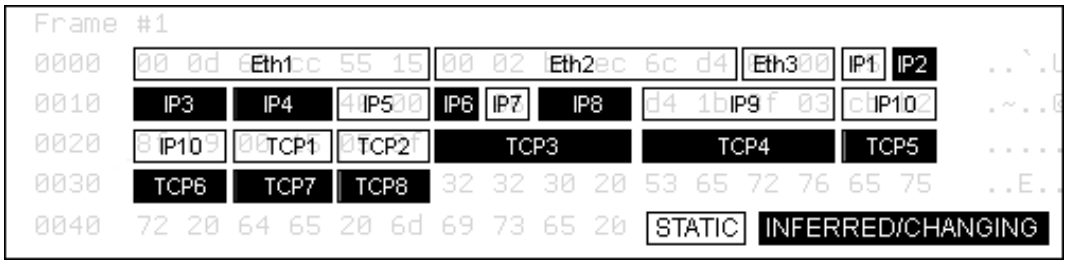


Figure 5.2: Header fields for the combined Ethernet/IP/TCP header

TCP5 : Data Offset, Reserved Bits and Flags (2 bytes). This field specifies the number of 4-octet words in the TCP header, indicating the start of the data. It is always a multiple of 4 octets, it can be deduced from the length of any options, and thus it is INFERRED. The Reserved Bits might be expected to be zero, but according to RFC 4413 this can no longer be assumed due to future-proofing. As for the Flags, they clearly vary during transmission: both the Reserved Bits and the Flags are therefore CHANGING.

TCP6 : Window Size (2 bytes). This may oscillate randomly between 0 and the receiver's window limit (for the connection), for which it is clearly CHANGING.

TCP7 : Header Checksum (2 bytes). This field is directly calculated from selected header bits, for which it is classified as INFERRED.

TCP8 : Urgent Pointer (2 bytes). If the URG flag is set, then the Urgent Pointer indicates the end of the urgent data and thus can point anywhere in the window. It may be set (and changing) over several segments. CHANGING.

Note that some header fields that were not tagged as STATIC do not appear to change in Figure 5.1 (e.g. **TCP6**: Window Size). This is due to the fact that only a few packets are shown, making the variability of some CHANGING fields not to appear clearly in the example. More precise analyses of header fields would naturally require longer packet samples.

In relation with table 5.2, Ethernet's 14 header fields are all STATIC, yielding the overall statistics of table 5.3 for our FTP negotiation.

	Ethernet/IPv6/TCP	**Ethernet/IPv4/TCP**
STATIC	54 **(73%)**	29.75 **(55%)**
INFERRED	2.5 (3%)	4.5 (8%)
CHANGING	17.5 (24%)	19.75 (37%)
TOTAL	74 (100%)	54 (100%)

Table 5.3: Summary of field categories for the Ethernet/IPv4/TCP example with FTP (last column). Ethernet/IPv6/TCP figures (first column) are presented for informative purposes.

5.1.3 Organization of this Chapter

The first part of this chapter defines the general framework for HERACLES, and introduces the main theoretical concepts and definitions associated to it. We then focus on the particular application of this framework to the common case of binary flows, over which HERACLES is said to operate in *hard* mode, in analogy with FEC techniques. Next, *soft* operation is analyzed, covering the cases where HERACLES performs processing over non-binary information flows, a common case in state-of-the-art physical layers. A particular case where HERACLES' soft mode can be of significant

importance for overall optimization is dealt with separately, followed by a section outlining the basic implementation issues related to practical operation. Finally, a synthetic discussion regarding HERACLES' strengths and weaknesses is followed by a comprehensive conclusion summarizing our main results.

5.2 Principle and General Framework

5.2.1 The Basics

The main idea behind HERACLES is the analysis of an incoming information flow upon reception in the search for occurrences of *known sequences*, characteristic of header redundancy. In order to do so, it sets up a cross-layer strategy allowing it to detect with great accuracy their original positions in the flow without triggering false alarms, even under very noisy conditions. Pinpointing of these known sequences paves the way for several enhancements to the overall system, that will be discussed along with the previously-mentioned detection strategy.

Definition of the Static Pattern (SP)

As shown in the previous section, headers of any information flow of organized packets (frames, encapsulated SNDUs, PDUs etc) share common STATIC fields such as those characterizing e.g. link or network-level source and destination addresses, ports or protocol types, not necessarily contiguous among them. We define the *Static Pattern* (SP) of a logical stream as the longest subset of STATIC fields that can be found in all headers of a given stream, and we denote by F its total number of symbols (either bits or bytes, depending on the context). A typical SP will be composed by series of contiguous STATIC bytes scattered among CHANGING and/or INFERRED header fields, with relative positions invariably reproduced in each packet header.

In the previous example, the SP is almost 30 bytes long (238 bits exactly, given that only 14 out of the 16 bits of the **IP5** field are STATIC, see Section 5.1.2). It is represented by the highlighted sequence in Figure 5.3, and its position in the packet is unambiguously referenced by pointing at its first byte, here in position 1 of the header.

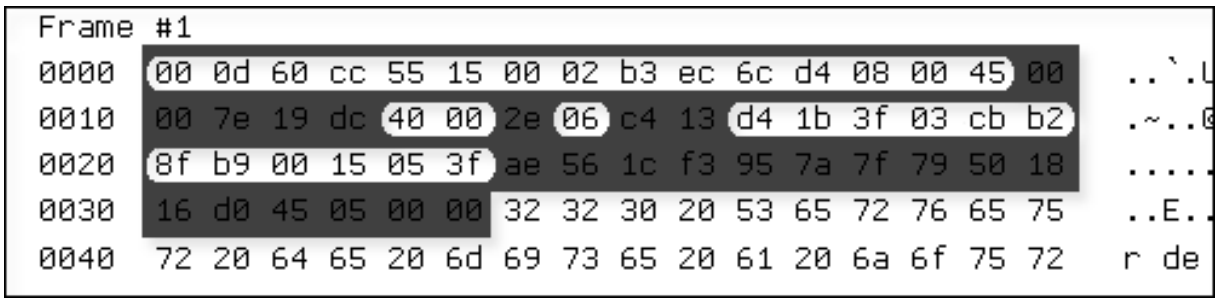


Figure 5.3: SP for the example of Section 5.1.2. The SP size is 238 bits long, and it is located in position 1.

Generic Model for HERACLES Operation

In order to analyze from a generic standpoint the functioning of HERACLES, consider the study case of Figure 5.4.

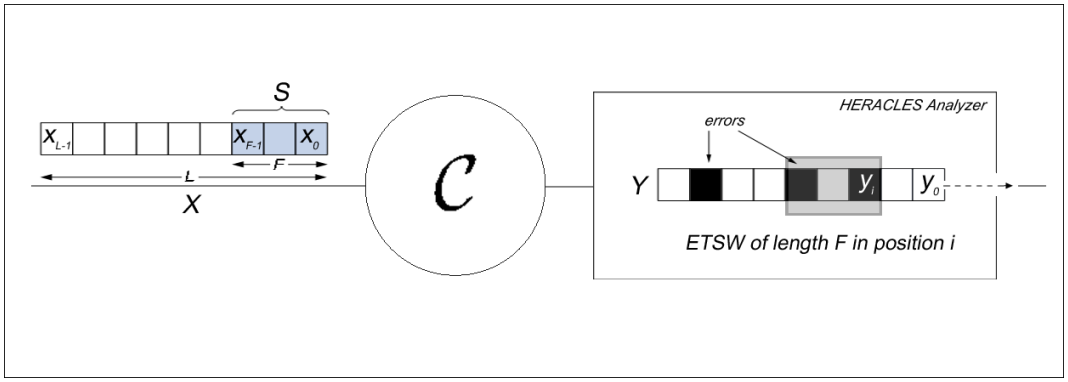


Figure 5.4: General transmission diagram for HERACLES operation. Transmission symbols are not represented in the classical network byte order: here, rightmost symbols are transmitted first.

Let $X = (x_i)_{i\in[0,L-1]}$ be a single packet of L symbols belonging to an information flow with known SP $S = (s_i)_{i\in[0,F-1]}$. S is assumed to be non-autocorrelated, reflecting the realistic fact that there is no structural dependance among the different STATIC fields whatsoever. The same assumption will be done for X, although chances are for this to be true to a lesser extent, depending on the specific payload data. Among the (x_i) symbols, $(L - F)$ represent varying header and/or payload data, whereas values — and relative positions in the header — of the F symbols belonging to the SP are well known. Although the (s_i) can be scattered among the (x_i), it will be supposed for the sake of clarity that they are contiguous and located at the beginning of X:

$$x_i = s_i \qquad \forall i \in [0, F-1] \tag{5.1}$$

Let $Y = (y_i)_{i\in[0,L-1]}$ be the received sequence after passage through a channel $\mathcal{C}$, feeding the HERACLES analyzer. The purpose of this block is to pinpoint the location of the original SP in the erroneous sequence Y: for this, it implements an *Error Tolerant Scanning Window* (ETSW), which scans methodically Y in the search for subsets of F symbols organized exactly as the SP, and resembling it *to a certain extent*. In order to quantify the extent to which a given symbol subset $Y_{/[i,i+F-1]} = (y_i, ..., y_{i+F-1})$ is similar to the known SP, we define for every possible position i in the received message a similitude measure z_i — in a metrics Ψ to define:

$$z_i = \Psi\left(S, Y_{/[i,i+F-1]}\right) \qquad \forall i \in [0, L-1] \tag{5.2}$$

Note that the last $(F - 1)$ values in the sequence of similitudes $Z = (z_i)_{i\in[0,L-1]}$ require the knowledge of symbols whose index appear to exceed the bounds of Y. Actually, this is by no means a problem: it precisely underlies the fact that the ETSW moves continuously over the received flow, using in practice some of the first symbols in the packet coming after X for this.

For every position i of the ETSW, a *detection* is triggered if the observed symbol subset $(y_i, ..., y_{i+F-1})$ is similar to the known SP, which can be expressed in terms of z_i exceeding a predefined detection

threshold η:

$$z_i \geq \eta \qquad \Leftrightarrow \quad \text{SP detection in position } i \tag{5.3}$$

5.2.2 Optimal Detection Strategy

Since the only SP is located in position $i = 0$, a detection triggered by the ETSW here is a *correct detection*. Every detection triggered for $i \neq 0$ constitutes a *false alarm*, meaning that the ETSW is fooled by a random sequence of symbols in the erroneous flow miming the structure of the SP. Of course, the ETSW is not able to make the distinction between a *correct detection* and a *false alarm*, for it does not to have any knowledge of the underlying structure of the flow. Clearly, the detection threshold η will play a major role in the relative occurrence of such events, for which its value should be carefully tuned.

The following paragraphs discuss the best way to leverage the overall situation by choosing an appropriate threshold detection η, leading to the most favorable configuration for detection. Ideally, the optimal threshold η_{opt} of an error-free flow should be set to the maximum possible value $\Psi(S, S)$, meaning that the ETSW should exclusively search for exact copies of the SP in the flow. For an erroneous flow, however, such setting would be too restrictive and lead to few or no detections, due to the almost certain corruption of the SPs. Unfortunately, lowering the detection threshold η has two opposite consequences: in addition to increasing the chances of correct SP detection — which has a positive effect on the system — it bears the risk of making the ETSW trigger an increasing number of false alarms. The optimal threshold will therefore require to be chosen with these two facts in mind, following a strategy designed accordingly.

Probability of Static Pattern Recovery PSR

An ETSW operating L scans over the received flow will unequivocally pinpoint the location of the unique SP if succeeds in detecting its presence in position $i = 0$ and in avoiding false alarms in the remaining $(L - 1)$ positions. At this point, we face a classical detection problem where design choices reflecting the desired behaviour of the system are to be made [131]. For instance, is it more important for our system to trigger a maximum number of correct detections than to avoid false alarms? Or, should we prevent false alarms from happening at any cost, even if that means missing some correct detections?

In order to define the bases of a generic mechanism, we have chosen to give equal importance to both correct detections and false alarms avoidance. We will therefore follow a detection strategy based on the maximization of a function we have called PSR, standing for Probability of Static Pattern Recovery and defined as follows:

$$PSR = \Pr\left[(z_0 \geq \eta) \bigcap_{i \neq 0} \overline{(z_i \geq \eta)}\right] \tag{5.4}$$

The outcome of every ETSW scan is independent from the others, and hence PSR factors into:

$$PSR = \Pr(z_0 \geq \eta) \cdot \prod_{i \neq 0} [1 - \Pr(z_i \geq \eta)] \tag{5.5}$$

The first term in the above equation constitutes the *Probability of Correct Detection* P_{cd}, which represents the probability that the distortion introduced by the channel $\mathcal{C}$ to the SP stays within the bounds defined by η. As for the remaining $(L-1)$ factors, only combinatory considerations dictate the presence of symbol sequences similar to the original SP — regardless of the precise scanned position i and noise affecting them. For this reason, all of them are identical to a unique *False Alarm Probability* P_{fa}.

Summarizing, PSR can be written as follows:

$$PSR = P_{cd} \cdot (1 - P_{fa})^{L-1} \tag{5.6}$$

with

$$\begin{cases} P_{cd} = \Pr(z_0 \geq \eta) \\ P_{fa} = \Pr(z_i \geq \eta) \quad \forall i \neq 0 \end{cases} \tag{5.7}$$

Note that other detection strategies could be followed according to specific system requirements. PSR provides however a fair basis for the generic study of HERACLES, and has the advantage of being conceptually handy from a mathematical standpoint.

PSR Maximization

Among the different variables affecting PSR, only the detection threshold η can be tuned by the analyzer. We define therefore η_{opt} as the optimal detection threshold maximizing PSR:

$$\eta_{opt}(X, S, \mathcal{C}) = \arg\max_{\eta > 0} PSR(\eta, X, S, \mathcal{C}) \tag{5.8}$$

Summary

By defining analytical expressions for P_{cd} and P_{fa} in a given system, the theoretical achievable Probability of Static Pattern Recovery (PSR) for a given flow can be defined as a function of the detection threshold η. From this point, provided that satisfactory estimations of the input parameters affecting P_{cd} and P_{fa} (and therefore PSR) are available, an iterative or recursive search of the optimal threshold value can be done. This tunes the HERACLES block to achieve the desired trade-off between accurate SP pinpointing and minimal false alarm triggering.

5.3 Hard Detection of Static Patterns

In order to illustrate the above considerations, we describe here what HERACLES operation looks like over a binary flow. In analogy with error correction techniques, we will qualify any processing done by HERACLES at bit — or byte — level as *hard*.

5.3.1 Preliminaries

Transmission Symbols

Suppose that both X and Y are byte flows, and that $\mathcal{C}$ is a binary symmetric channel (BSC) with crossover probability ε at bit level. ETSW scans could be done for every bit position, but this would lead to unnecessary processing in the general case. Indeed, if properly chosen, SP starts are always aligned with 8-bit boundaries in the flow, making 7 scans pointless out of 8 if done on a per-bit basis.

ETSW scans can therefore be done byte-by-byte in the incoming sequence Y, and both L and F can be expressed in bytes.

Similitude Metrics

The Hamming distance is an appropriate tool for comparing binary sequences, given that likeness between two binary sequences can be measured in terms of *low* Hamming distances. In addition, the framework employing this metrics classically used for hard decoding in binary error correction is rather convenient for expressing the extent of similarity between two sequences. Classically, a received (noisy) word V can be decoded into a known codeword U if V lies within in the decoding sphere of U and radius t.

Using a similar approach, we postulate that an observed byte subset $(y_i, ..., y_{i+F-1})$ under the ETSW triggers a detection if it lies in the sphere of radius η bits centered on S. In other terms, only byte sequences $(y_i, ..., y_{i+F-1})$ differing in at most η *bit* positions with the SP will trigger detections. Although the Hamming distance calculation and the expression of η could be done in bytes, preferring the bit level allows for better granularity and accuracy.

5.3.2 PSR Expression for Hard Detection

Under the previous assumptions, the probability of correct detection P_{cd} is the probability that at most η bit errors occur among the $8F$ bits under the ETSW in its first scan, with no special conditions on their positions. Classical combinatory analysis yields to:

$$P_{cd} = \sum_{k=0}^{\eta} \binom{8F}{k} \varepsilon^k (1-\varepsilon)^{8F-k} \tag{5.9}$$

In any of the $(L-1)$ remaining byte positions, a false alarm will be triggered if the scanned byte subset is inside the sphere of radius η bits centered on S. In total, 2^{8F} possible bit sequences can be done out of F bytes, among which the ones presenting η bit errors or less — in any position — lie in the sphere of radius η bits centered on S:

$$P_{fa} = \frac{1}{2^{8F}} \sum_{j=0}^{\eta} \binom{8F}{j} \tag{5.10}$$

The previous equations lead to an analytical expression for PSR using equation (5.6):

$$PSR = \sum_{k=0}^{\eta} \binom{8F}{k} \varepsilon^k (1-\varepsilon)^{8F-k} \left[1 - \frac{1}{2^{8F}} \sum_{j=0}^{\eta} \binom{8F}{j} \right]^{L-1} \tag{5.11}$$

5.3.3 PSR Study

The general variations of PSR as a function of η are shown in Figure 5.5. In this precise example, a 100-byte sequence with SP size $F = 16$ bytes was analyzed under extremely noisy conditions $\varepsilon = 10^{-1}$ (1 bit out of 10 in error). For small values of η the PSR is very low, meaning that the

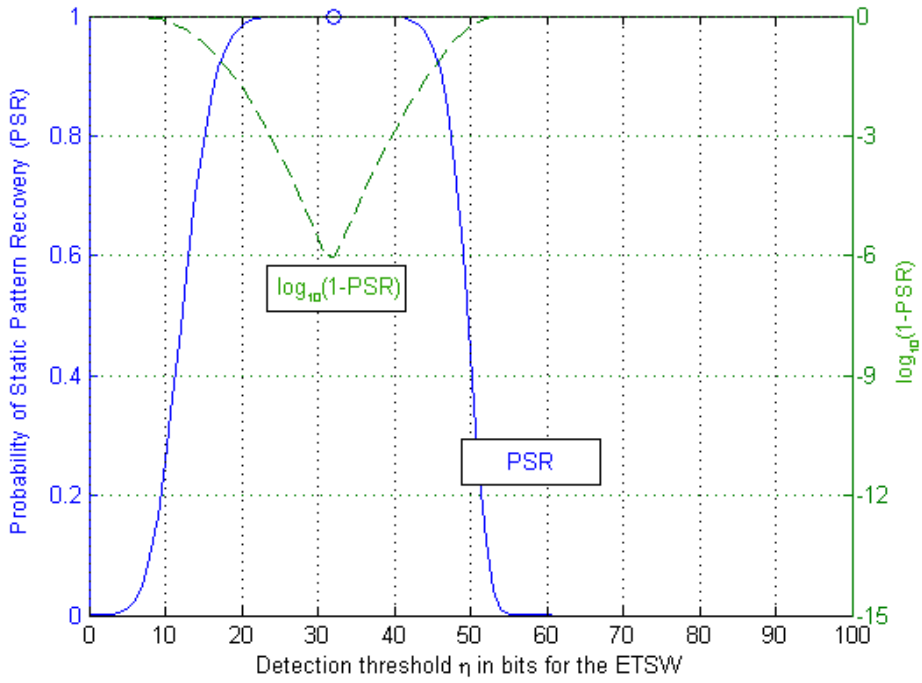


Figure 5.5: PSR as a function of η for $F = 16$ bytes and $\varepsilon = 10^{-1}$. The dashed line (right-side scale) represents the logarithmic distance between PSR and one, i.e. $log_{10}(1 - PSR)$.

ETSW has very limited chances of accurately detecting the SP if the flexibility of the search is not increased. Next, PSR rises fast with increasing η and remains very close to unity for η in a given interval: in this *plateau* all values of η produce high values for PSR, allowing for excellent SP pinpointing. Finally, PSR drops abruptly for large values of η, which is easily explained by the increased number of false alarms that start occurring if the detection threshold is chosen too large.

In order to have a better understanding of the phenomena occurring at the PSR plateau, a logarithmic zoom of $(1 - PSR)$ is presented by the dashed line in Figure 5.5. The zoom allows to see that for this precise case, choosing $\eta_{opt} = 32$ leads to the highest mathematical PSR ($PSR = 1 - 10^{-6} = 0.999999$). Interestingly enough, all values for η in the plateau around η_{opt} lead also to PSR close to one, implying excellent detection capabilities even though η is not totally

optimal. From a practical point of view, this might happen e.g. when the estimation of the input parameters L and ε needed to calculate η_{opt} has not been very accurate, giving excellent robustness to the mechanism's detection strategy.

5.3.4 Performances and Applications

Flow Delineation for Corrupted and Non-Corrupted Flows

SP pinpointing in an information flow can be done with great accuracy if the strategy based on PSR maximization is followed. Given that successfully locating successive SPs in a flow directly leads to determination of packets lengths and boundaries, HERACLES could be used for pure *delineation* (sometimes called packet synchronization). This seems of particular interest for link and adaptation layers, in replacement or complement of classical delineation/framing methods, at no overhead at all. In particular, HERACLES provides robust delineation *even for erroneous packet flows*. Indeed, state-of-the-art delineation techniques all rely on data integrity, by the use of sensitive header information such as payload pointers and length fields, or synchronization sequences and sliding hashes. HERACLES decouples the delineation problem from the issues regarding data integrity, and therefore opens a new range of possibilities. Take for instance the reduction or replacement of synchronization sequences (see e.g. [132], [133] and [134]) or its use in protocol stacks with error-tolerant applications, where erroneous packets wanting to climb the protocol stack are unfortunately erased due to header corruption and/or synchronization losses. In order to quantify the accuracy of this delineation technique, let's analyze P_{cd} and P_{fa} under different noise conditions.

$\underline{\varepsilon = 0:}$ $P_{cd} = 1$ and $P_{fa} = 2^{-8F}$. In other words, all SPs are detected *without exception*, and false alarms *never* occur, given that for F around 4 or 5 bytes, P_{fa} is already below 10^{-10} (which is explained by the combinatory explosion caused from the moment that 32 or 40 bits constitute the SP). For F around 20 bytes (IPv4), P_{fa} is below 10^{-49}! If implemented at a layer benefiting from QEF conditions, delineation with HERACLES is therefore extremely accurate.

$\underline{\varepsilon \neq 0:}$ Suppose now that we want to delineate an *erroneous* flow of encapsulated packets with SP size, say, 6 bytes long. Figures 5.6 and 5.7 show PSR variations for packets $L = 100$ and $L = 1500$ bytes long[2], with SP sizes F up to 16 bytes under noise conditions ε between 10^{-4} and 10^{-1}. For all of them, $PSR \approx P_{cd} \approx 1$, meaning that all original SP locations are perfectly identified by HERACLES. What about false alarms? A series expansion to first order of equation (5.6) for $P_{cd} \approx 1$ and $P_{fa} \ll 1$ leads to a fair approximation of the probability of false alarm P_{fa}:

$$P_{fa} \approx \frac{1 - PSR}{L - 1} \tag{5.12}$$

P_{fa} has been plotted with PSR in Figures 5.6 and 5.7 as well. They clearly show that just like in the error-free case, P_{fa} decreases very fast with increasing SP size. Graphically, the above approximation even allows to determine the minimum required SP size achieving a given target P_{fa}. Just like in the error-free case, the use of HERACLES with classical SP sizes of few tenths of bytes makes false alarms extremely improbable events, not likely to ever occur during the lifetime of the system.

[2]These figures assume that proper estimates for L and ε exist, so that η_{opt} has been found and PSR has been maximized accordingly. See Section 5.6.2 for more information on this precise point.

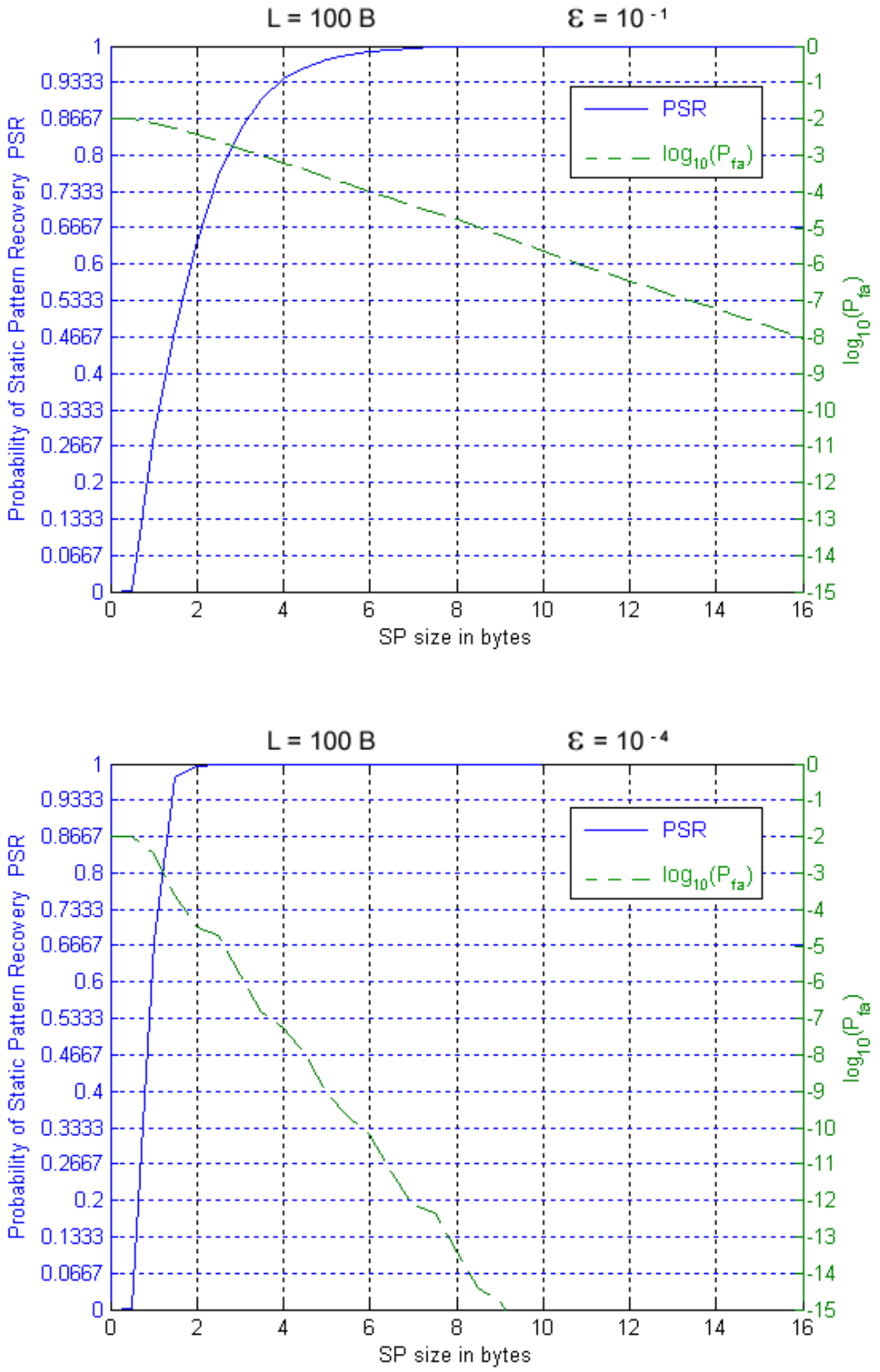


Figure 5.6: PSR and delineation accuracy: P_{fa} vs. SP size F for $L = 100$ bytes under $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-4}$.

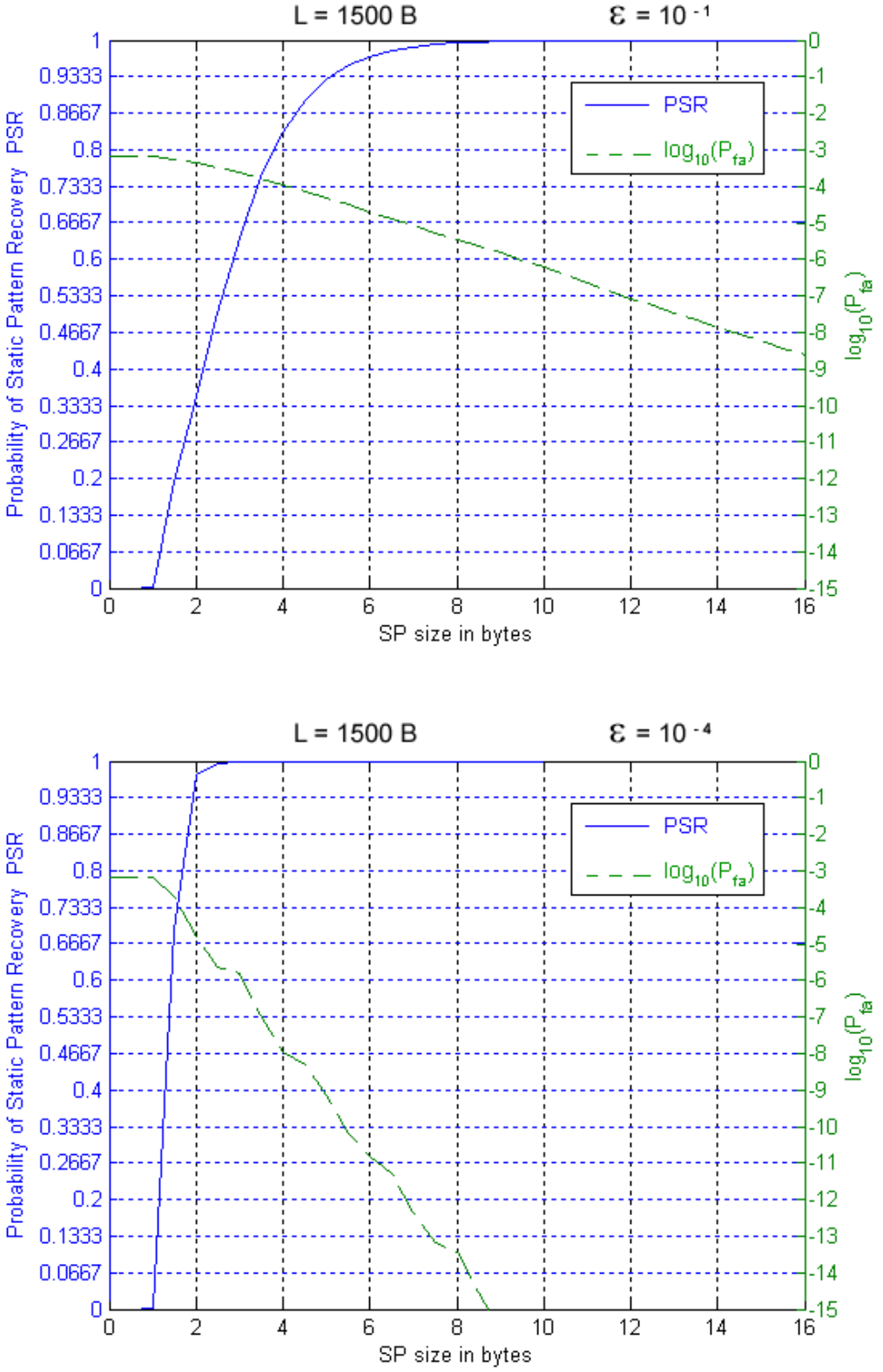


Figure 5.7: PSR and delineation accuracy: P_{fa} vs. SP size F for $L = 1500$ bytes under $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-4}$.

Error Correction

Given that SP pinpointing can be done with great accuracy, bytes subsets leading to detection could be replaced by hard copies of the searched SP, directly lowering the BER perceived by the following stage or layer in the communications chain. In every packet an average of $PSR \times F$ bytes

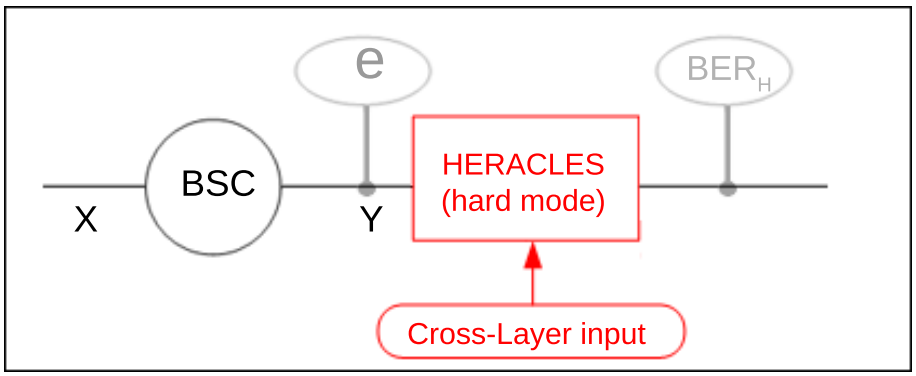


Figure 5.8: Basic diagram for BER reduction with HERACLES in hard mode.

out of L can be corrected by HERACLES. The achieved BER_H after this operation, referring to notations of Figure 5.8, can therefore be estimated by:

$$BER_H \approx \varepsilon \left(1 - PSR\frac{F}{L}\right) \tag{5.13}$$

Such BER decrease may not seem very important at first glance, although for small packets with long headers (e.g. VoIP) it could reach values close to 50% and more. However, as it will be shown in Section 5.5, such reduction can have a huge impact on the overall system if properly exploited.

Finally, note that no special assumption has been done on the layer at which the mechanism can be used, given that the BSC abstraction may cover any layer subset of the protocol stack. This allows for HERACLES being deployed transparently at any level of the receiver's decoding chain, up to its uppermost layers.

The Algorithm

A practical algorithm for implementing flow delineation and/or error correction with HERACLES in a real system is presented in Figure 5.9.

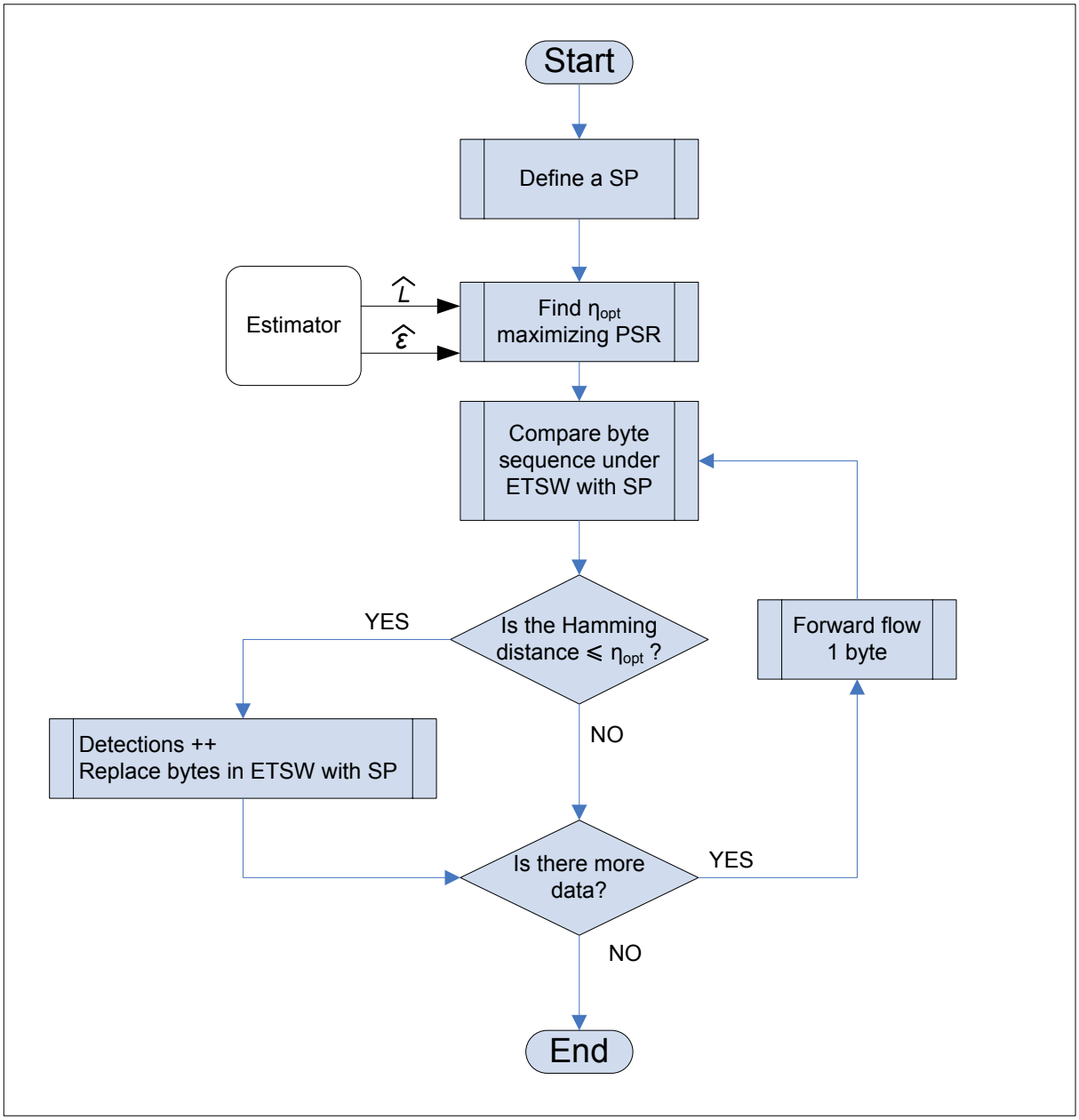


Figure 5.9: Practical algorithm for flow delineation and/or error correction with HERACLES in hard mode.

5.4 Soft Detection of Static Patterns

5.4.1 Preliminaries

On Hard and Soft Values

Now suppose that Y is a series of real — or *soft* — values, such as the raw flow of information produced by "soft output" devices like soft demodulators or SISO (Soft-In-Soft-Out) FEC decoders. In these cases and assuming equally likely input bits, soft values are therefore the Log-Likelihood Ratios (LLR) of the input bit values, as illustrated in Figure 5.10.

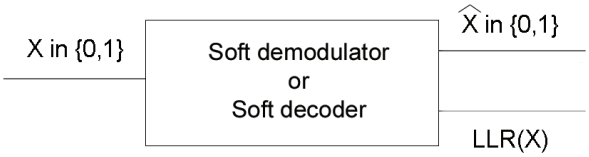


Figure 5.10: Example of a transmission block with soft output.

$$LLR(X) = \log\left(\frac{\Pr\left(X=0|\hat{X}\right)}{\Pr\left(X=1|\hat{X}\right)}\right) \tag{5.14}$$

Contrary to the previous section where bytes were the transmission symbols, soft values are defined here for every transmitted bit. This basically implies that ETSW scans have to be done on a per-bit basis and that F and L are to be expressed in bits.

This said, let $X = (x_i)_{i\in[0,L-1]}$ and $S = (s_i)_{i\in[0,F-1]}$ be the soft representations of our initial *binary* sequences, using the canonic hard-to-soft isomorphism:

$$\left|\begin{array}{l}\{0,1\} \to \{-1,1\}_{\subset\mathbb{R}} \\ \quad 0 \mapsto 1 \\ \quad 1 \mapsto -1\end{array}\right. \tag{5.15}$$

The Channel

In order to develop a working example of soft detection, let $\mathcal{C}$ be an AWGN channel characterized by the real gaussian noise vector $\Phi = (\varphi_i)_{i\in[0,L-1]}$ affecting the transmitted bits:

$$\forall i \in [0, L-1] \quad \varphi_i \sim \mathcal{N}(0,\sigma^2) \qquad \text{with} \quad \sigma^2 = N_0/2 \tag{5.16}$$

Assuming soft demodulation, the received noisy message $Y = (y_i)_{i\in[0,L-1]}$ can be written as follows:

$$y_i = a\cdot x_i + b\cdot\varphi_i \quad \forall i \in [0, L-1] \tag{5.17}$$

where $\{a, b\} \in \mathbb{R}^2$ characterize the specific modulation and soft decoding type.

Similitude Metrics

When dealing with sequences of real values, the discrete cross-correlation — commonly used for pattern recognition in digital signal processing — is a natural tool for likeness analysis. As a reminder, for two real soft sequences $U = (u_i)_{i\in\mathbb{Z}}$ and $V = (v_i)_{i\in\mathbb{Z}}$, the cross-correlation vector denoted $\mathcal{R}_{UV} = U \star V$ is defined by:

$$(\mathcal{R}_{UV})_p = \sum_{n\in\mathbb{Z}} u_n v_{p+n} \quad \text{for} \quad p \in \mathbb{N} \tag{5.18}$$

The higher the similarity between U and V is, the greater the cross-correlation. For this, U and V being "similar to a given extent" can be straightforwardly expressed in terms of $(\mathcal{R}_{UV})_p$ exceeding a predefined detection threshold $\eta \in \mathbb{R}$.

5.4.2 Correlation Analysis

We have chosen the discrete cross-correlation as our similitude metrics Ψ. In order to analyze the degree of similarity between every subset of F contiguous symbols in Y contained in the ETSW and the known SP, we calculate the associated similitude measure $Z = (z_i)_{i\in[0,L-1]}$ from equation (5.2):

$$z_i = S \star \left(Y_{/[i,i+F-1]}\right) \quad \forall i \in [0, L-1] \tag{5.19}$$

Since both sequences S and $\left(Y_{/[i,i+F-1]}\right)$ have F elements in practice, the index of the correlation sum only spans the range $[0, F-1]$ instead of $\mathbb{N}$. Developing the above expression hence yields:

$$z_i = \delta_i + e_i \qquad \text{with} \begin{cases} \delta_i = a \sum_{n=0}^{F-1} s_n x_{i+n} \\ e_i = b \sum_{n=0}^{F-1} s_n \varphi_{i+n} \end{cases} \tag{5.20}$$

Study of e_i

Since the term e_i in equation (5.20) is a linear combination of F independent white gaussian noises following normal distributions $\mathcal{N}(0, \sigma^2)$, e_i follows a normal distribution itself, with mean and variance determined by the same linear coefficients (s_n).

- Mean: $E[e_i] = b \sum_{n=0}^{F-1} s_n E[\varphi_{i+n}] = 0$
- Variance: $\text{var}(e_i) = b^2 \sum_{n=0}^{F-1} (s_n)^2 \,\text{var}(\varphi_{i+n}) = b^2 F \sigma^2$, given that $(s_n)^2 = 1 \; \forall n \in [0, F-1]$.

Summarizing, all the e_i follow the normal distribution $\mathcal{N}\left(0, Fb^2\sigma^2\right)$.

Study of δ_i

Here, two cases arise according to the position i:

- For $i = 0$, given that $x_n = s_n$ by condition (5.1) and that $(s_n)^2 = 1\ \forall n \in [0, F-1]$, we simply have $\delta_0 = Fa$. This means that a correlation peak of intensity Fa occurs when the ETSW is over the SP. Note that δ_0 can also be seen as a normally distributed random variable with mean Fa and no dispersion at all:

$$\delta_0 \sim \mathcal{N}(Fa, 0) \tag{5.21}$$

- For $i \neq 0$, given that S is not autocorrelated, δ_i is the sum of F independent random variables following Rademacher distributions. From the Central Limit Theorem, $(\delta_i)_{i\neq 0}$ can be approached by the normal distribution:

$$(\delta_i)_{i\neq 0} \sim \mathcal{N}\left(\sum_{n=0}^{F-1} E\left[as_n x_{i+n}\right], \sum_{n=0}^{F-1} \mathrm{var}\left(as_n x_{i+n}\right)\right) \tag{5.22}$$

with:

$$\begin{cases} \sum\limits_{n=0}^{F-1} E\left[as_n x_{i+n}\right] = \sum\limits_{n=0}^{F-1} as_n E\left[x_{i+n}\right] = 0 \\ \sum\limits_{n=0}^{F-1} \mathrm{var}\left(as_n x_{i+n}\right) = a^2 \sum\limits_{n=0}^{F-1} (s_n)^2 \mathrm{var}\left(x_{i+n}\right) = Fa^2 \end{cases} \tag{5.23}$$

In conclusion, all the $(\delta_i)_{i\neq 0}$ follow normal distributions $\mathcal{N}(0, Fa^2)$.

Extensive studies exist on Rademacher sums such as [135] and [136], which could be used to refine the results given by the Central Limit Theorem in order to get more complex and accurate considerations for δ_i. However, the normal approximation of their convergence provides a fair and convenient basis for our study.

Summary

From the above considerations, it is clear that all elements in Z can be expressed as sums of normally distributed random variables. Putting all together, we finally have:

$$z_i = \begin{cases} \delta_0 + e_0 \sim \mathcal{N}\left(\mu_0, \sigma_0^2\right) & \text{for} \quad i = 0 \\ \delta_i + e_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right) & \text{for} \quad i \neq 0 \end{cases} \tag{5.24}$$

with

$$\begin{cases} \mu_0 = Fa \quad \text{and} \quad \sigma_0^2 = Fb^2\sigma^2 \\ \mu_i = 0 \qquad \text{and} \quad \sigma_i^2 = F\left(a^2 + b^2\sigma^2\right) \quad \text{for} \quad i \neq 0 \end{cases} \tag{5.25}$$

Summarizing, a maximum of correlation occurs for $i = 0$: a peak of intensity Fa modulated by gaussian noise with variance σ_0^2 is characteristic of the presence of a SP in the soft sequence at the ETSW. For any other position $i \neq 0$, the result of the cross-correlation of the SP and the examined soft sequence can be seen as pure white gaussian noise with variance σ_i^2.

5.4.3 PSR Expression for Soft Detection

Now that similitude observations have been classified as draw outcomes of normally distributed random variables, analytical expressions for P_{cd} and P_{fa} to be used in equation (5.6) can be easily derived.

When in position $i = 0$, a correct detection occurs when the observed distance is above the detection threshold. However, such event is to be considered as a false alarm for every other position $i \neq 0$. It follows that:

$$P_{cd}(\eta) = \frac{1}{\sigma_0^2\sqrt{2\pi}} \int_{\eta}^{+\infty} \exp\left(-\frac{1}{2}\frac{(t-\mu_0)^2}{\sigma_0^2}\right) dt \tag{5.26}$$

$$P_{fa}(\eta) = \frac{1}{\sigma_i^2\sqrt{2\pi}} \int_{\eta}^{+\infty} \exp\left(-\frac{1}{2}\frac{(t-\mu_i)^2}{\sigma_i^2}\right) dt \tag{5.27}$$

Hence the final expression for PSR in the soft detection case:

$$PSR = \left[\frac{1}{\sigma_0^2\sqrt{2\pi}} \int_{\eta}^{+\infty} \exp\left(-\frac{1}{2}\frac{(t-\mu_0)^2}{\sigma_0^2}\right) dt\right] \cdot \left[1 - \frac{1}{\sigma_i^2\sqrt{2\pi}} \int_{\eta}^{+\infty} \exp\left(-\frac{1}{2}\frac{(t-\mu_i)^2}{\sigma_i^2}\right) dt\right]^{L-1} \tag{5.28}$$

5.4.4 PSR Study

Just like for the hard detection case, η_{opt} can be found by the criteria of equation (5.8).

PSR variations in the soft detection case are identical to those shown in Figure 5.5. A PSR plateau exists for η in a given interval around the optimal value, meaning that in practice all η values in this interval lead to excellent SP pinpointing.

In order to provide a better understanding of the soft detection process, Figure 5.11 shows what soft detection for HERACLES based on correlation observation looks like, using two SP sizes $F = 48$ bits and $F = 128$ bits under the same noise conditions. In these examples, two soft sequences of 100 packets each were analyzed by HERACLES after passage through an AWGN channel inducing $BER = 10^{-1}$ with soft QPSK demodulation.

Despite the extremely high error rate (1 bit out of 10 in error), correlation peaks can be easily observed every 100 bytes for both cases, revealing the locations of the original SPs. Under $BER = 10^{-1}$, almost unequivocal SP pinpointing in the flow can be achieved providing that the SP is at least 48 bits (6 bytes) long.

5.4.5 Performances and Applications

Soft Flow Delineation

Note that the uppermost case in Figure 5.11 ($F = 16$ bytes, $BER = 10^{-1}$) corresponds to the experimental conditions used for Figure 5.5, and that in both scenarios $PSR \simeq 1 - 10^{-6}$. Simulation and theoretical results confirm indeed that PSR reaches comparable values for both hard and soft detection scenarios under similar conditions, showing that soft SP recovery can be also used for flow delineation with the performances quantified in Section 5.3.4. However, there is an important difference: soft values are dealt with in physical layers only, meaning that *flow delineation in packets of any level could be done right from the physical layer* with the right SP choice. As before, this delineation technique might help complementing or partially offloading the delineation function classically implemented in adaptation layers, with no added overhead at all.

Error Correction

Just like in the hard detection scenario, the accuracy achieved in the SP pinpointing process legitimately allows for modification of the transmission symbols leading to detections with SP information, providing a small amount of error correction to the overall flow. Given that soft values represent confidence degrees on the a priori value of a given bit, correction of the i^{th} soft value in the matching subset could consist e.g. in assigning to it a high absolute value affected with the sign of s_i, as illustrated in Figure 5.12.

Such action cannot be quantified in itself in terms of reduced BER, for it should be matched with a hard decoder as shown in Figure 5.13 in order to make a comparison at bit level. Following the same considerations as those of Section 5.3.4, a very simple hard detector such as an integrator delivers a binary flow with BER_H satisfying equation (5.13).

Of course, such correction is localized — it only affects headers — and bounded. However, as the next section will show, when matched with more sophisticated soft-input devices such as SISO FEC decoders, important synergies can be achieved.

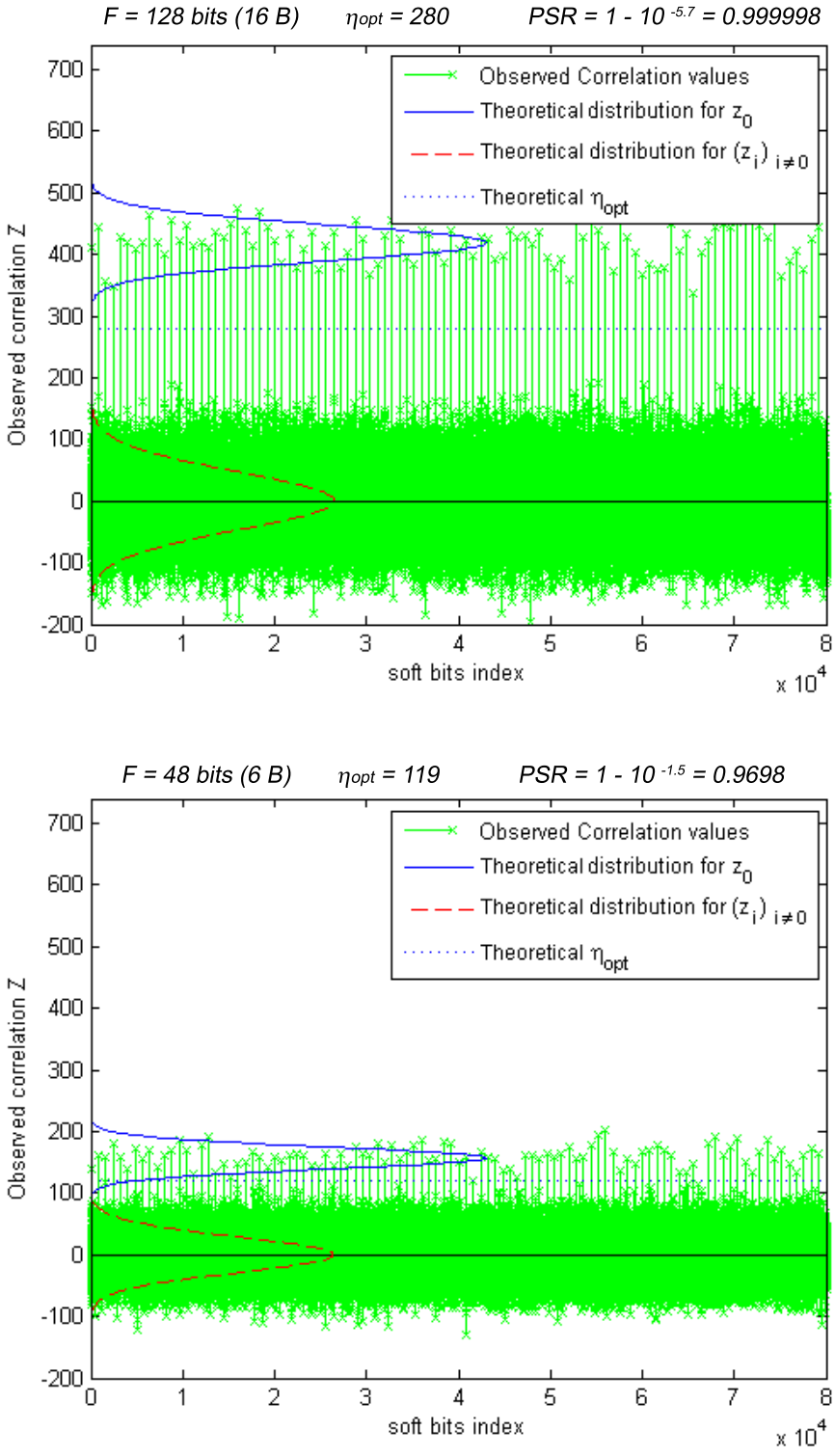


Figure 5.11: Experimental and theoretical correlations (z_i) for a series of 100-byte long packets with SP sizes $F = 128$ bits (top) and $F = 48$ bits (bottom) over an AWGN channel ($BER = 10^{-1}$) with soft QPSK demodulation. The scale for the Gaussian distributions has been magnified (not shown).

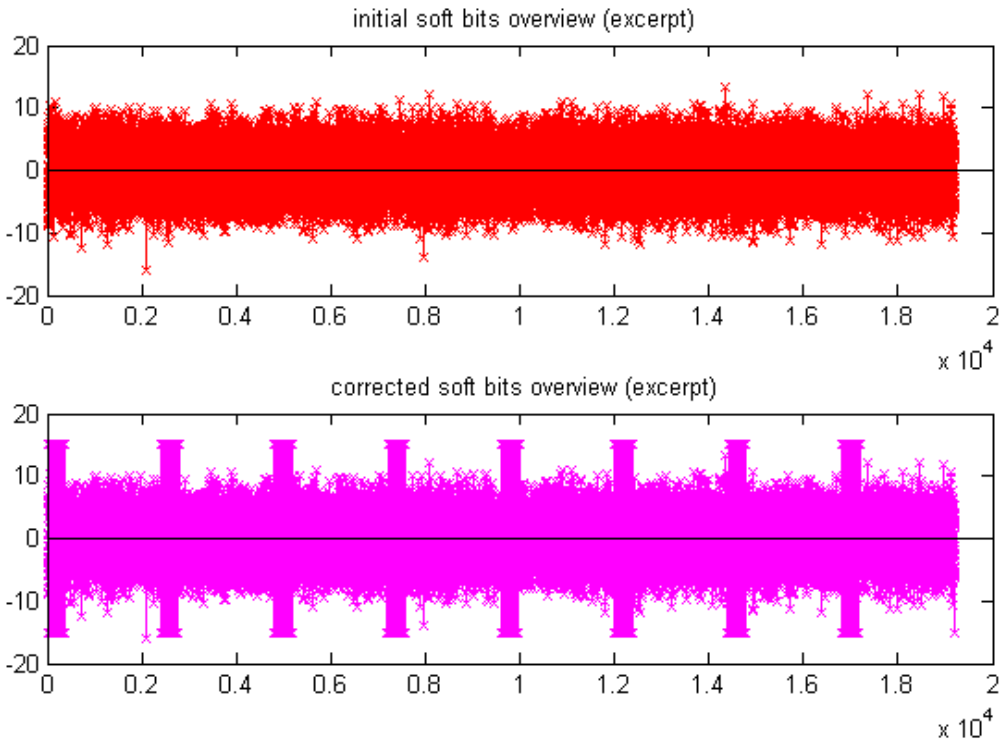


Figure 5.12: Example of replacement of soft values with known SP info in symbol subsets leading to detections. Here, every soft value leading to detection is given an absolute value of 15.

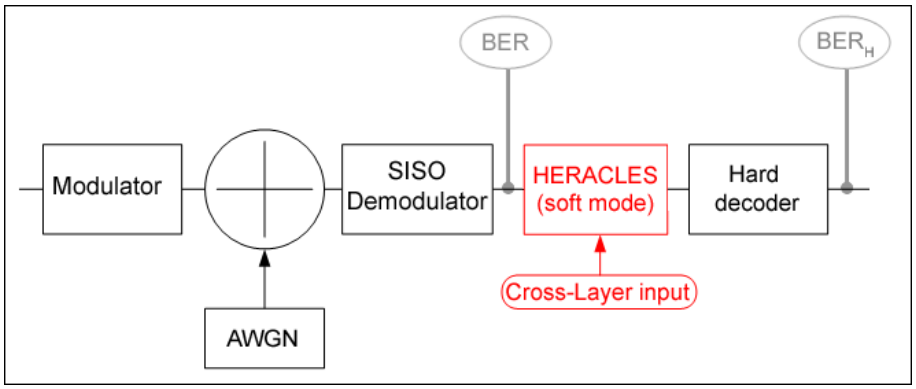


Figure 5.13: Basic diagram for BER reduction with HERACLES in soft mode.

The Algorithm

The use of HERACLES in its soft mode does not affect the basic structure of the algorithm presented in its hard mode. Figure 5.14 shows a soft version of this algorithm, in which only minor (mostly terminology) changes are done.

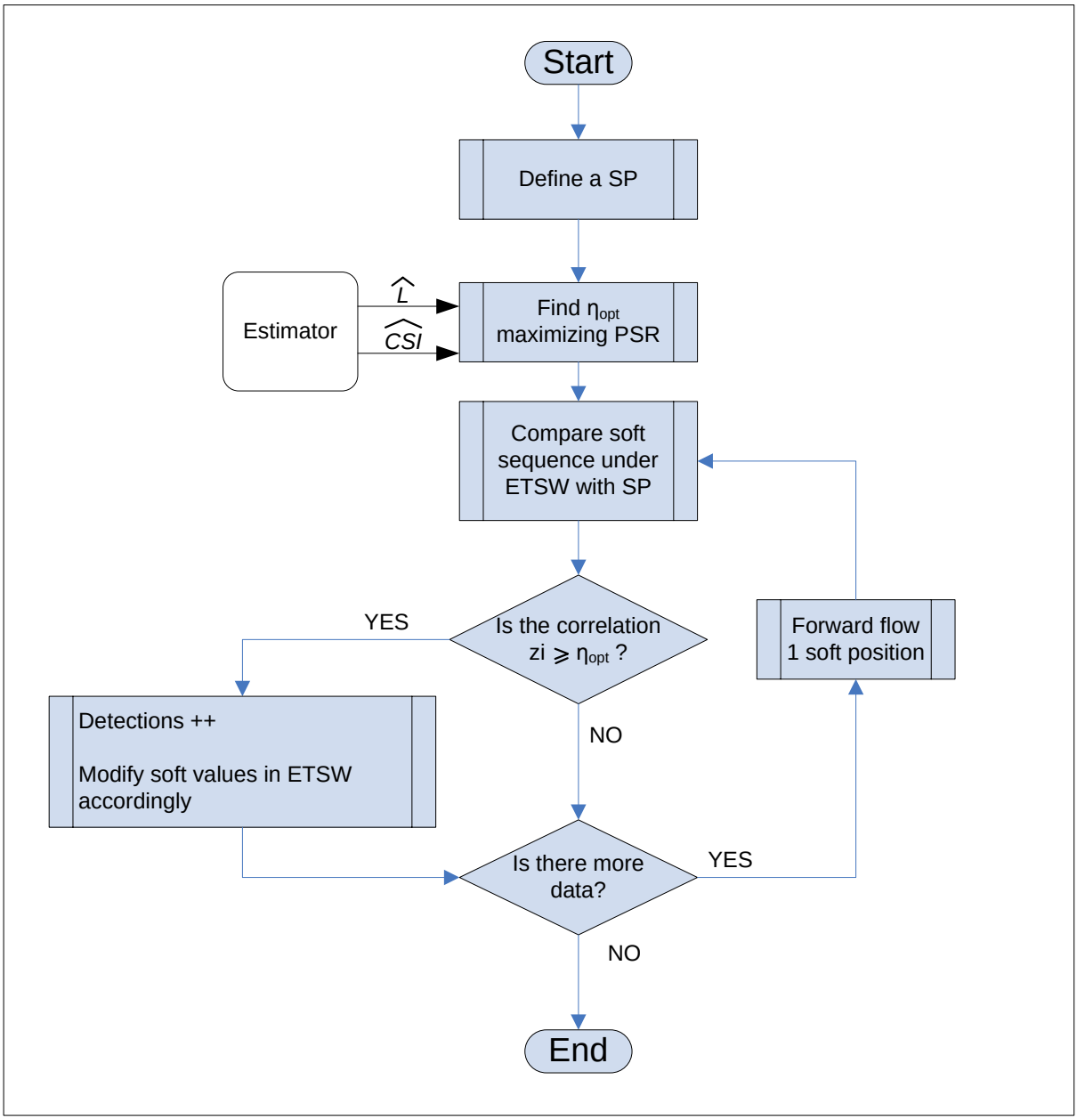


Figure 5.14: Practical algorithm for flow delineation and/or error correction in HERACLES' soft mode

5.5 Combined Use of Soft HERACLES Detection and FEC

5.5.1 Motivation

The previous sections showed that HERACLES allows for some errors to be detected and corrected at different layers, both in its hard and soft detection modes. This raises the question of the best way to exploit this handy feature in the sake of overall optimization, by examining HERACLES' potential integration at different points in the communication chain. Clearly, one of the best ways to exploit HERACLES' capabilities would be to find potential synergies with the available error correction functions such as checksums, ARQ, CRCs or error correction codes. Given that the FEC subsystem provides the most important capability in this sense, it seems appropriate to examine how HERACLES and a FEC codec could work together.

One could imagine several possible hybrid FEC/HERACLES configurations in hard or soft modes, of course, but a clear lead derives from this fact: state-of-the-art error correcting codes such as Turbo codes [137] are soft input devices, and they present very steep slopes in the BER_{output} vs. BER_{input} domain. This basically means that a small reduction of their soft input's noise has the potential to generate huge decoding gains. For this reason, one of the most promising configurations consists in using HERACLES' soft mode as a preliminary error correcting device, intended to feed the FEC decoder with a — slightly — cleaner flow than the one delivered by the soft demodulator.

5.5.2 General Scenario

The purpose of this section is to evaluate to which extent HERACLES can impact the behaviour of a soft-input FEC decoder, as it was previously suggested. For this, we compare the performances of the two transmission systems shown in Figure 5.15: (A) being a classical scheme, and (B) an HERACLES-enhanced chain.

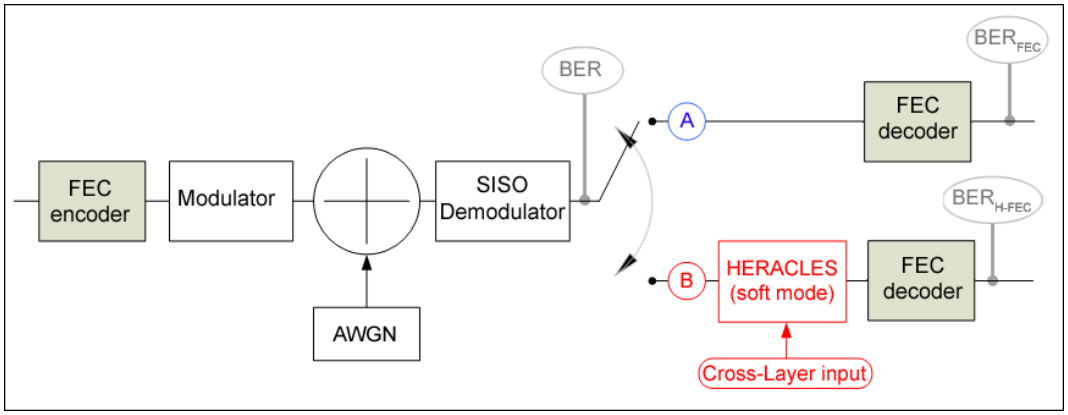


Figure 5.15: Study case for comparing a classical transmission chain (A) and an HERACLES-enhanced system (B).

An important fact to be noted is that the flow feeding the HERACLES block in chain (B) has been FEC-encoded, potentially jeopardizing the success of an SP research in the HERACLES block by

disrupting the SP inner structure. For this reason, only systematic codes — i.e. codes where the input data are embedded in the encoded output — can be used here. Among systematic codes, two cases are to be considered:

- *Systematic block codes:* Parity symbols are appended to an entire coding block, usually at its end. SP structure is unaffected by this operation in the general case, which makes FEC coding transparent for HERACLES operation. Note however that block segmentation prior to coding may result in those SPs located on block boundaries being fragmented and therefore harder to exploit in the context of our study.
- *Systematic interleaved codes:* Parity and data symbols are intertwined in the coded flow according to a predefined order. Although not that straightforward, SP recovery can still be done by several ways, e.g. by ignoring parity symbols, or by looking for coded instances of the SP instead of the SP itself [132][133][134].

5.5.3 Application Case With the 3GPP Turbo Code for UMTS

Two series of 100-byte long packets with SP sizes $F = 160$ and $F = 320$ bits were sent through chains (A) and (B) in Figure 5.15, using the Turbo code specified in the 3GPP standard for the European 3G Universal Mobile Telecommunications System (UMTS) [138][139] and used in DVB-SH [41][42] with code rate $r = 1/3$ and interleaver size $K = 12282$ bits (Figure 5.16). In this systematic code, every data bit x_k generates 2 parity bits p_1^k and p_2^k, which are finally interleaved according to the sequence $x_1, p_1^1, p_2^1, x_2, p_1^2, p_2^2 \ldots$

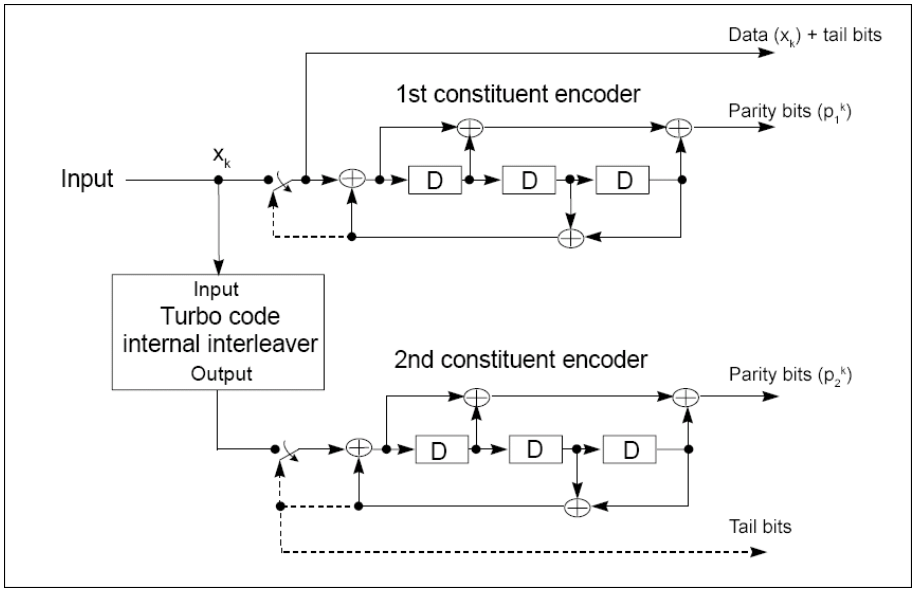


Figure 5.16: Structure of the 3GPP systematic Turbo Encoder for UMTS and DVB-SH . Dotted lines apply for tail bits only, used in treillis termination (source: ETSI).

As a first step for this analysis, soft SP search was performed in the coded sequences by looking only at data bits and ignoring parity symbols. Finally, soft values leading to detections after the HERACLES block in chain (B) were modified as suggested in Section 5.4.5. Final BER and PER

values were measured for both chains (A) and (B) under various noise conditions, with results presented in Figures 5.17 and 5.18.

Both cases show important reductions in BER and PER after FEC decoding when using HERACLES, attaining up to 1 dB for $F = 320$ bits. They show in particular that the localized and reduced error correction provided by HERACLES around the SPs has the potential to trigger an avalanche of corrections at the FEC decoder that spreads all over the packets, improving substantially the quality of the overall decoding.

5.6 Practical Considerations

The previous sections dealt mostly with the theoretical aspects of HERACLES, and the presented examples/figures assumed an ideal and smooth functioning of the mechanism. If implemented in real systems, however, practical engineering issues arise.

Among the implementation challenges affecting the mechanism, the cross-layer access and use of the information required for the mechanism is one of the most interesting ones. Indeed, after identifying a working SP, the system has to provide valid estimates for the input BER and L in order for HERACLES to maximize PSR using equation (5.6): this requires HERACLES being able to query and/or retrieve such estimations from the other layers of the protocol stack; upper layers for the SP and — classically — physical layer for the noise estimation.

This section presents a list of possibilities that a real system implementing HERACLES could consider for identification of a working SP and the practical estimation of these parameters.

5.6.1 SP Determination

A key issue to solve for practical operation is the definition, memory loading and duration management of a working SP at the receiver side. Clearly, this is a critical step since it conditions the behaviour and performances of the whole mechanism.

The definition of a working SP should follow 3 basic steps, different in difficulty and straightforwardness of application under different contexts.

1. *Scope definition*. Section 5.2.1 defined the SP as the longest subset of STATIC fields that can be found in all the headers of a stream. Truth is, by subtracting some fields from the longest possible SP, it becomes possible to change the scope of the streams affected by HERACLES operation. Concretely, a receiver (an end node or an aggregating host indifferently) could focus on a single and specific flow, or it could process all the incoming flows from a specific source by including or subtracting IP or MAC addresses to the SP. Note finally that nothing prevents a receiver from defining several simultaneous SPs and therefore from launching several simultaneous HERACLES instantiations.
2. *Protocol stack identification and SP mask selection*. Once the flows to be processed by HERACLES are selected, the system should unequivocally identify the protocol stack used

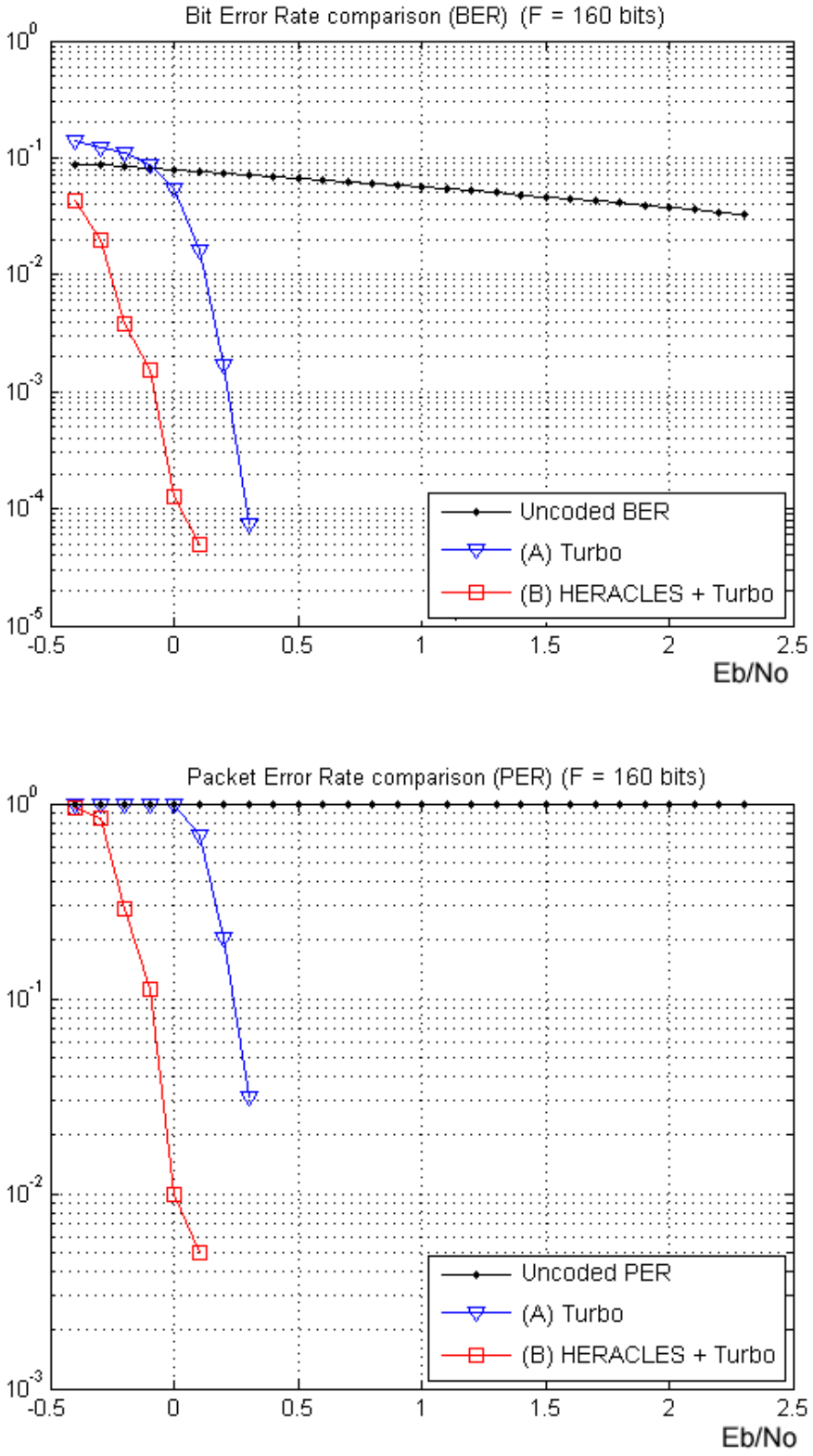


Figure 5.17: BER and PER figures for the 3GPP Turbo code ($r = 1/3, K = 12282$) used in UMTS: alone (A) and HERACLES-enhanced (B), $F = 160$ bits (20 bytes) and $L = 100$ bytes.

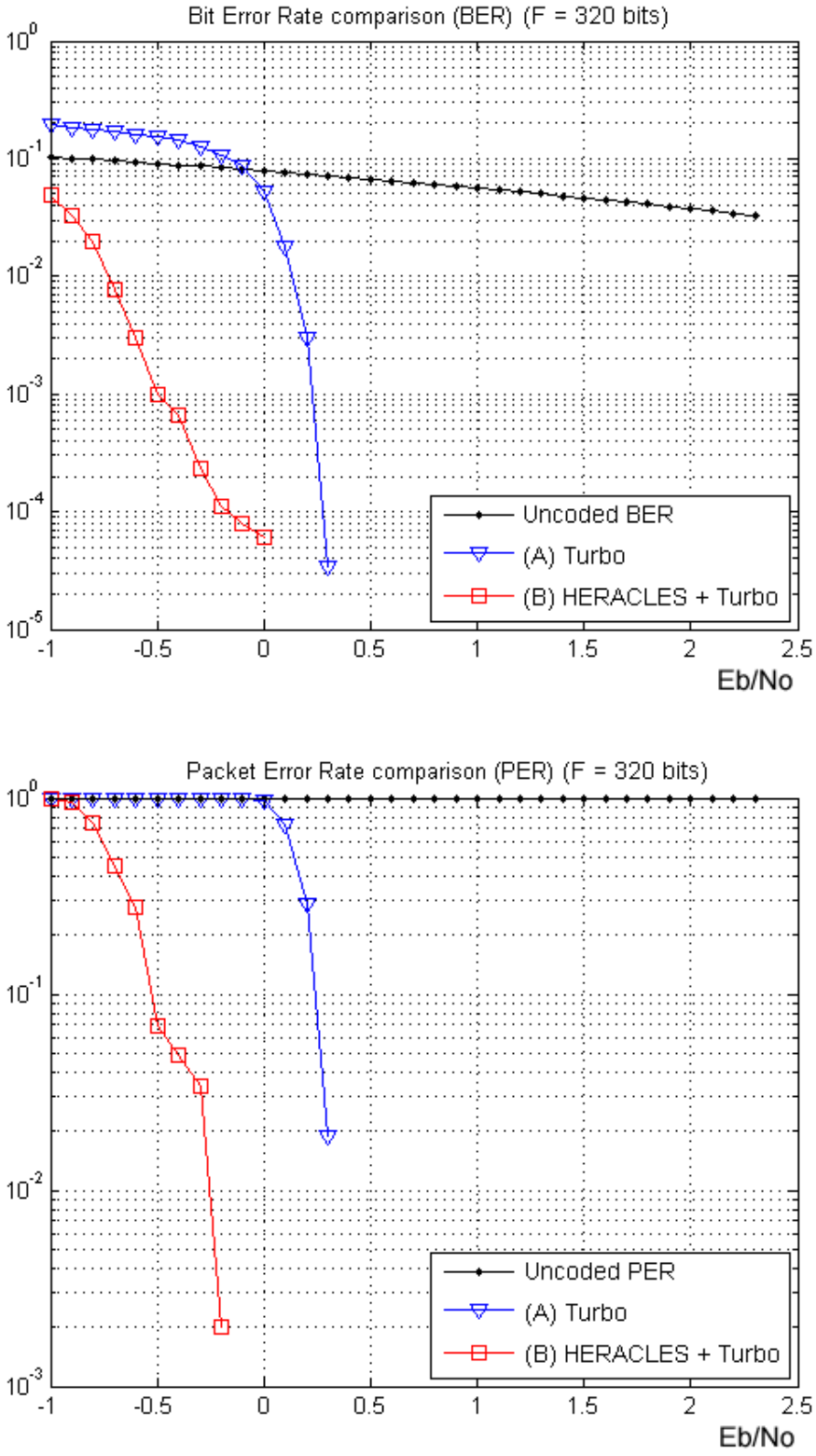


Figure 5.18: BER and PER figures for the 3GPP Turbo code ($r = 1/3, K = 12282$) used in UMTS: alone (A) and HERACLES-enhanced (B), $F = 320$ bits (40 bytes) and $L = 100$ bytes.

by them. Possible ways to do this are the use of a cross-layer function, the exploitation of statistical data at the receiver side or a return channel. This is a mandatory and very important step, since every protocol stack can be characterized by the particular organization or distribution of its STATIC fields within the header. In a possible implementation, such information could be exported e.g. under the form of a binary mask to be used by the ETSW during its scans, or be preloaded in a table. The delivered mask should of course integrate the scope choices done previously by the system by toggling on/off the use of some of the SP fields.

3. *Affectation of SP fields*. Once the inner structure of the SP has been loaded, the system can proceed to the affectation of the specific STATIC values that constitute the SP. This might be a tricky part: although a receiver has an inherent knowledge of many SP fields (e.g. its own IP and/or MAC addresses), it ignores in principle the contents of the SP fields related to the transmitter (e.g. network identifiers). A very simple cross-layer mechanism delivering this information could be imagined here in order to determine the remaining values, doubled by a dynamic cache/learning function.

Finally, there is the timing issue regarding the validity of a working SP. Clearly, a SP should be defined for a limited period of time, matching as close as possible the lifetime of its corresponding stream(s). Again, cross-layer exchanges could help HERACLES determine the most adequate timing policies, for instance if a dialog with those layers introducing timeouts such as TCP or ULE is allowed. Unfortunately, no universal solutions exist for this particular issue: only case by case analyses of HERACLES integration in specific communications chains allow precise considerations in this sense.

5.6.2 Input Parameters Estimation and Sensitivity

Below are presented some techniques that can provide good estimates for both L and the input BER in the author's opinion.

Estimation of L

- Given that the term $(1 - P_{cd})$ in equation (5.6) is so small in the general case, L has a marginal influence on PSR. Hence existing statistics on the used protocol stack can give a first rough working estimate for L during the SP lifetime.
- A "hard value" for the estimation of L can be set and used during the lifetime of the system. Although this method is less precise than the previous one, the small influence this parameter has on PSR justifies this approach. If this method is to be used, hard-setting a large value for L provides safer results than a small one, since PSR is degraded with very large L. As an example, the maximum transmission unit (MTU) of the layer to which the packets belong could be chosen.
- Almost real-time estimation of L can be done as well, with a simple function keeping track of previously recovered packets and their observed length (e.g. by analyzing the number of

symbols between two successive SP detections), recorded in a dedicated memory and re-injected periodically to the ETSW. The length information can also be accessible through a dedicated cross-layer mechanism, able to retrieve it from other layers and pass it to the ETSW.

As mathematical and simulation results confirm, the packet size distribution itself has no influence on the performances of HERACLES. Indeed, ETSW scans are independent and done over the continuous information flow, totally ignoring packets boundaries.

BER Estimation at the Input of the HERACLES Block

- From "bottom to top", through the use of Channel State Information (CSI) when available.
- From "top to bottom", exploiting the observed Packet Error Rate at the upper layers. A cross-layer function could observe the statistics on discarded packets and infer the channel BER using a posteriori estimates. Such information could be completed e.g. by the use of a return channel, able to provide more precise information about the successful processing of upper layer packets e.g. by counting TCP ACKs and NACKs.
- The FEC decoder could infer the channel BER from statistics on the number of errors it corrects in every decoding block, and pass this information to HERACLES. Following a similar idea, HERACLES' input BER can be roughly estimated with the number of erroneous bits corrected by the ETSW during replacement of the analyzed byte sequences with the used SP.

Given that the input BER plays a more important role in PSR than L, the system should pay special attention to channel's noise estimation. In addition, BER variations in time are expected to be much more important than for L, for which its estimation is inherently more delicate. Note that many techniques not listed here can be employed to estimate the BER at the input of HERACLES, and that several works and ongoing research on the more general topic of CSI estimation exist such as [140],[141] and [142].

Sensitivity to Estimation Errors

In order to study the sensitivity of HERACLES to the potential estimation errors that may occur, a first possibility is to test HERACLES operation under "noisy" estimated parameters. However, given the wide spectra of values that the input BER and L can take, the sensitivity of HERACLES to these parameters can be studied directly from the analysis of PSR variations.

As stated previously, errors on the estimated value of L has very low influence on PSR. In the examples of Figures 5.6 and 5.7 it can be seen that under $\varepsilon = 10^{-1}$, changing L from 100 to 1500 bytes barely increases the required SP size to guarantee a given PSR.

An error of an order of magnitude on the BER estimation has different consequences. For the information flow of Figure 5.6, reaching $PSR = 1 - 10^{-6}$ under $\varepsilon = 10^{-4}$ requires $F = 5$ bytes, but $F = 16$ bytes or more are needed under $\varepsilon = 10^{-1}$ in order to achieve the same PSR target.

5.7 Discussion

5.7.1 Limitations of the HERACLES Solution

HERACLES can be used as long as the SP's inner structure is intact, so that the ETSW has chances to extract it out of an information flow. Unfortunately, there are several factors that can alter its structure in the transmission chain and therefore jeopardize the operation of HERACLES. The importance of such impairments mostly depend on the number of layers separating the mechanism from those to which the SP belong, which directly conditions all the processing at the receiver before the HERACLES block. The SP is likely to be made out of network, transport and application layer header fields, and HERACLES is likely to be implemented in the lower layers. As a rule of thumb, the lower the layer at which HERACLES is implemented at the receiver, the higher will be the chances of SP structure being altered due to the transmitter's layers successive data processing. HERACLES implemented in higher layers is likely to be unaffected by these considerations.

A non-exhaustive list of factors that can impinge HERACLES operation by altering of modifying SP structure is presented here.

Header compression. When using header compression, header redundancy is suppressed or reduced. Under such circumstances, HERACLES cannot use upper layers redundancy. However, the mechanism could provide some limited functionality at least if implemented below, using a secondary SP built upon STATIC fields from layers not concerned by compression. Good choices are adaptation/encapsulation layer fields when available, since those are rarely compressed and contain addressing or next protocol information, very likely to be repeated from packet to packet. In Section 5.7.3, "MPE-HERACLES supporting ROHC" refers to this configuration.

Sub-network or FEC Framing. Physical and/or link layer fragmentation into FEC blocks or SNDUs with predefined sizes is common in digital communications systems. Unfortunately cases may occur when block or SNDU fragmentation occurs over a SP instance, making it locally unrecognizable by HERACLES operating over the segmented (non yet reassembled) flow. A palliative measure consists in setting up a dedicated cross-layer channel between HERACLES and the reassembly function of the receiver, tagging block or SNDU edges as zones to be dealt with carefully.

Interleaving. HERACLES is currently unable to recognize a SP in an interleaved flow. Indeed, physical layer interleaving modifies bit order to serve particular purposes (e.g burst errors spreading), and therefore entangles SP bits as well.

Non-systematic FEC. In order to exploit the potential synergies of HERACLES with a FEC decoder, the mechanism should be able to recognize SP occurrences in a *coded* flow. As stated in Section 5.5, this basically means that HERACLES cannot be used directly over flows coded with non-systematic FEC schemes such as convolutional codes.

5.7.2 Advantages of the HERACLES Solution

Besides its **flow delineation** and/or **error correction** capabilities, the mechanism introduced in this chapter presents a series of interesting advantages that make it a suitable tool for enhancing digital communications systems.

First, HERACLES proposes a **novel** use for header redundancy, especially suited to error-prone communications systems. Its design relies on an innovative **cross-layer** approach making simultaneous use of both upper and lower layers information, which together define the best possible configuration for its use.

In addition, given that the HERACLES block does not require any protocol, global architecture or data modification prior to SP search, it can be implemented in a **non-intrusive** and **protocol-compliant** manner by a receiver at almost **any level of the protocol stack**. This means that the HERACLES block can be seen as a simple "plug-in" that could be turned on or deactivated at will without requiring any modification at the transmitter side, to which the use of HERACLES is completely transparent. Furthermore, since HERACLES can be implemented at various levels of the transmission chain, it has the potential to be deployed both in **intermediary and end nodes** according to specific system requirements. In addition, it exhibits a very **limited linear complexity**, given that the similitude calculations (Hamming distance and correlation) required for every ETSW scan are extremely simple and fast linear operations.

Finally, note that its range of application covers **not only satellite**-based communications networks, but also any kind of communication system making use of packetized information flows.

5.7.3 Aimed Protocol Stacks

Non-exhaustive protocol stacks where HERACLES could be used include:

IP/TCP. IP/TCP - based stacks support the most popular and widely deployed applications such as HTML/Web browsing, FTP transactions etc. HERACLES could be used at any level of the IP/TCP stack, preferably in hard mode. Of course, enhanced HERACLES operation can be achieved with the inclusion of STATIC fields of protocols both below and above the IP and TCP layers, such as Ethernet for instance (see Figure 5.1).

IP/UDP. Its popularity and use is similar to IP/TCP, and statistics on its STATIC fields when used with RTP are available in Table 5.1. In particular, it can be seen that a SP with size $F = 20$ bytes can be chosen for a single flow of IPv4/UDP/RTP packets. Mathematical results show that using $F = 20$ bytes for any packet size allows having $PSR \geq 1 - 10^{-7.4}$ for *any* ε value below 10^{-1}, meaning that even under such noisy conditions, all SPs of an IPv4/UDP/RTP flow can be found unequivocally. For IPv6/UDP/RTP, $PSR \geq 1 - 10^{-15}$ for *any* ε value below 10^{-1}, thanks to the almost doubled SP size ($F = 44$ bytes) when compared to IPv4/UDP/RTP.

MPE with IP/UDP/RTP or IP/TCP. If some MPE fields are included in the definition of the SP, ETSW scans could be done at SNDU level. ULE is somehow less suited for this purpose because of its simpler header semantics, and especially the non-mandatory inclusion of the

destination MAC address in its basic header. If only the 6-byte MAC destination address of the MPE header is taken into account, the extended SP for IPv4/UDP/RTP becomes $F = 26$ bytes long. In this case, PSR can be raised up to $1 - 10^{-10}$ for *any* ε value below 10^{-1}. For MPE/IPv6/RTP/UDP, an extended $F = 50$ bytes makes PSR exceed the precision of our numerical tools, although it can be extrapolated to be greater than $1 - 10^{-20}$ for any ε value below 10^{-1}! HERACLES implemented at an adaptation layer and using STATIC fields from it could decisively contribute to the delineation process for instance.

MPE-HERACLES supporting ROHC. HERACLES can also give support for header compression schemes such as ROHC under extremely bad BER figures. Indeed, error-tolerant applications such as speech coding, which could allow frame error rates (FER) up to 1% [143] very seldom employ header compression schemes. Indeed, context re-synchronization is strongly affected by high post-FEC bit error rates: as a benchmark, ROHC performs poorly for $BER \geq 10^{-5}$ [129]. One could therefore imagine a case where ROHC packets encapsulated over MPE could be left with some resilient errors, and be delineated using MPEG2 and MPE fields (e.g. MAC address) *and* the context identifier byte(s) of ROHC as SP in a first step. This would allow recovering the context identifier field on the packets of the ROHC flow, allowing the compression scheme to work properly. Precise mathematical and simulations results are still to be done for this particular case in order to evaluate its pertinence.

ATM ATM-based transmission schemes such as DVB-RCS or the ADSL could also benefit from HERACLES.

5.7.4 Extensions and Future Work

Extending the Header Redundancy Concept

A natural direction to examine for further improvement is the general use of *redundancy*, regardless its form and location in the flow. Indeed, redundancy is not only limited to the existence of STATIC header fields in a packetized flow, but can appear under many different ways in a message, both within packet headers and data as well.

For instance, one could imagine the existence of a function able to use the knowledge of the way the INFERRED fields change in headers within the flow, in order to complement the correction capabilities already brought by the STATIC part of the header. A practical example of this is given by Length fields: since SP recovery allows excellent flow delineation, any corrupted Length field can be recovered by counting the symbols separating two successive SP detections. For other INFERRED fields such as counters or flags, advanced studies on their variability exist for several known protocol stacks, since they are the basis over which general header compression schemes are built on.

In the general case, redundancy appears within the transmitted data as well, provided that it complies with the syntax rules of well identified languages (English, HTML, etc) or applications. Take the English language for example, whose redundancy patterns and structures have been widely studied from the beginning of communications theory [7][144]. Video flows or HTML pages have also redundancy patterns in the forms of information tables or tags that, if properly identified and used, have the potential to contribute to the overall effort of future HERACLES-like mechanisms.

In this regard, DUDE-like concepts for identifying structural data patterns in information flows are certainly a good starting point for further reflection.

Better FEC/HERACLES Integration

Section 5.5 discussed how placing the HERACLES block before the FEC decoder allowed for the latter to perform better. Future work could explore the possibilities of enhancing this HERACLES/FEC synergy e.g. by defining a joint decoding stage. Indeed, it seems interesting to analyze to which extent HERACLES can directly influence the decoding algorithm of the FEC subsystem, instead of just being a preliminary stage. Reciprocally, we have identified potential enhancements in the detection process of the HERACLES block with the use of FEC information: from this standpoint, the definition of a combined stage makes sense.

Explore Further HERACLES Integration Possibilities

The framework defined in this chapter is general enough to serve as a basis for integrating HERACLES at different levels of the protocol stack. Section 5.5 focalized on a particular application case with a strong potential for system enhancement, but much work is still to be done in order to identify the best configurations for HERACLES use in a complete system, and especially, the best HERACLES configuration for every particular context. A thorough exploration of the different possibilities offered by the mechanism has the potential to redefine global error control strategies, with the possibility of achieving more balanced and effective policies for the sake of overall optimization.

Enriching HERACLES through Plug-ins

In addition to its core mechanisms, several "plug-ins" could be imagined to add new functionalities to HERACLES and enhance its overall performances. For instance, ETSW scans could be coupled with integrity checks (e.g. CRCs or other hash mechanisms) calculated from the desired SP. One could then imagine a trial-and-error series of bit flips performed on selected sequences, in order to attempt reconstitution of the desired information.

5.8 Conclusions

5.8.1 Summary

This chapter introduced HERACLES, a new intra-host cross-layer mechanism bringing delineation and error correction capabilities to an information flow at no extra bandwidth cost. Implemented at the receiver of a communications chain, it exploits the natural redundancy existing among headers in packets belonging to an information flow. HERACLES searches and locates known predefined sequences in the received message accurately, which paves the way for enhanced flow framing and recovery.

Mathematical and simulation results showed that in order to maximize the chances of flow recovery, the mechanism must capture and/or estimate information coming from the adjacent layers, such as CSI or the average packet length. HERACLES presents excellent robustness and synchronization capabilities: even under extremely noisy conditions, most (if not all) packets can be correctly located within the flow and if wanted, most of header errors can be corrected.

Bit error rate reductions reaching up to 50% in realistic protocol stacks can be observed with the raw use of this mechanism. Furthermore, when matched with particular soft-input FEC decoders, HERACLES has the potential to topple the behaviour of full systems working at the limits of their nominal domains, bringing practical gains up to ~ 1 dB.

5.8.2 Patents and Related Work

The framework presented in this chapter has become a promising way to enhance bandwidth-limited and error-prone communications systems by making use of innovative cross-layer techniques. Although it is at its early definition stages, the HERACLES framework first described in a technical report [20] has been covered by two patent applications filled in Q4-2007 and Q1-2008 by Thales Alenia Space [26][27] and has recently been submitted [145] for publication.

Recent research have started exploring the cross-layer possibilities offered by upper layers redundancy for synchronization or error control purposes. Among them, the author identified [146], [147], [148] and especially the Discrete Universal Denoising (DUDE) framework [123][149][150] as the most closely related to the presented work. Although these References deal with concepts similar to the ones used in this study, their differences in scope are important and can be considered as complementary works.

Several patents making use of scanning windows for synchronization purposes in physical and link layers exist. Among them we can quote packets/cells recovery mechanisms for ATM [151], or MPEG2 [152], which make both use of direct header information (checksums and synchronization bytes) in order to achieve resynchronization. All of them assume that seen from the receiver the system behaves as a packet erasure channel (bad packets after FEC are all discarded by ad-hoc mechanisms such as CRCs); so that header information — when available — can be fully trusted. So far, we have not found any evidence of patents claiming error-tolerant delineation or error correction techniques using native header redundancy.

Chapter 6

Conclusions

6.1 Summary

The work presented in this document was motivated by the desire to achieve better and more efficient error control for satellite links by means of cross-layer initiatives.

We introduced cross-layer design in Chapter 2 as a new research area in wireless communications that advocates enhanced collaboration between layers beyond the scope of classical layered architectures. Section 2.2.2 showed that in wireless contexts with QoS provisioning and in particular in satellite cases, such architectures present shortcomings directly translated in costly transmission inefficiencies (redundancies, resource waste, excessive delays and so on). We then showed that cross-layer initiatives *do* have the potential to optimize some aspects of wireless communications to a certain extent. We reviewed some of the most important proposals in this sense up to now in Section 2.3, and identified 3 major axes for cross-layer enhancements in the literature. These were congestion & rate control, QoS and smart resource allocation. Finally, we weighted the strengths and weaknesses of this new approach. We concluded that cross-layer design bears a real potential for system optimization, but that over-excessive confidence in cross-layer design should be avoided for the sake of long-term architecture sustainability.

Chapter 3 assessed the way error control is managed in the lower layers of DVB satellites, by studying how FEC and adaptation layer CRCs interact to provide error-free data to the network layer. Analyses of the error patterns at the output of a DVB-S FEC subsystem in Section 3.3 showed that the outer Reed-Solomon decoder is aware of the vast majority of frame errors occurring upon decoding and SNDU reassembly, and that resilient or undetectable errors account for less than 10^{-5} (or 0.001%) of the times a CRC check fails in MPE and ULE. We also reminded that this information is unfortunately unknown by CRCs, who have to find all the errors on their own after thorough analysis of every single SNDU. This suggested that the bandwidth-consuming task of the SNDU integrity check could be at least partially offloaded to the FEC subsystem, at no extra-cost and safely. For this, we proposed an intra-host cross-layer mechanism authorizing the FEC decoder to share its decoding information with the adaptation layer, using either an in-band or out-of-band signaling procedure. Next, we extended these ideas to DVB-S2 in Section 3.4, and determined that the aforementioned ratio dropped to 10^{-8} (or 0.000001%) for 17 out of its 21 decoding

configurations. GSE's choice not to append a CRC to every single SNDU — saving an average of 4 bytes per packet — was then justified under the lights of DVB-S2's enhanced FEC scheme and longer bearers sizes. This leads to non-negligible bandwidth savings for future DVB-S2 links, quantified in around 10% for small packets — whose growing proportion account today for more than 40% of the exchanged packets in the Internet.

Next, Chapter 4 presented the work related to the definition and standardization in Europe of the new adaptation layer for IP over DVB-S2, the Generic Stream Encapsulation (GSE) protocol. In particular, it described how work on this topic started almost from scratch at the beginning of this thesis, by means of discussions at the IETF based on an Internet Draft published in August 2005 and informal discussions with DVB and ESA through Thales Alenia Space. The rationale and motivations for a new adaptation layer were exposed, stressing in particular the reasons why the existing mechanisms used for DVB-S were not adapted to the DVB-S2 context. We then moved to GSE's header formats and main characteristics in Section 4.5, and showed how the results obtained in Chapter 3 motivated GSE's design choices for error control. Given that it allows full use of DVB-S2's adaptive physical layer and cross-layer friendly features, we identified GSE's flexibility to transmit payload fragments in non-consecutive link-layer frames as its most important characteristic, relegating its overhead figures to a second plane. In the end, some words on its current status and future adaptation to other DVB radio layers concluded this chapter.

Finally, Chapter 5 introduced an innovative and standard-independent cross-layer framework for error control called HERACLES, which stands for Header Redundancy Assisted Cross-Layered Error Suppression. HERACLES was entirely developed in this thesis, and consists on a series of cross-layer functions implemented at the receiver to bring *overhead-free* delineation[1] and error correction capabilities to packetized — and possibly erroneous — information flows. Instead of relying on added control information (such as synchronization pilots or parity symbols) at the transmitter, HERACLES exploits the natural redundancy existing among headers of packets belonging to an information flow. It utilizes data's structural redundancy to assess packet positions in the bit stream, and performs header bit corrections if desired, based on carefully weighted success probabilities. In particular, it was shown in Section 5.5 that when HERACLES' output is directed to the input of an appropriate FEC decoder, important synergies can be triggered. HERACLES basically behaves as an inner error correction code, providing the FEC decoder with a somewhat cleaner version of the data flow it would have received without it. In those common cases where the FEC decoder is working just below the limits of its functional domain, the small correction brought by HERACLES has the potential to make the FEC subsystem toggle from a non-decoding to a full decoding state. Computer simulations show that in many cases, enhancements up to 1 dB can be observed under realistic system configurations using common TCP/IP stacks.

6.2 Future Directions

With the aforementioned contributions, this thesis showed that quantifiable enhancements could be done in the area of error control by means of cross-layer mechanisms. Moreover, it opened research leads that may pave the way for more and better results in this area in the years to come.

[1] Also known as packet synchronization or flow delimitation.

Below are summarized the most important directions for future research in the author's opinion, already exposed in Chapters 3 and 5.

6.2.1 Future Developments for GSE

Undoubtedly, the evolution of wireless and satellite networks will allow or require richer functionalities to be added to GSE in the years to come. The native mechanism that could be used for this purpose is described in Section 4.5.4. On top of the existing extension headers already defined, other potential uses of this mechanism include support for compression, QoS-signalling and performance monitoring. For instance, an extension to the encapsulation is being considered to provide confidentiality (encryption) and optional source authentication.

GSE Adaptation to other DVB Radio Layers

In particular, additional functions may be provided in the near future to adapt GSE to other DVB radio layers, either existing e.g. DVB-SH or to come, such as evolutions of the DVB-H, DVB-RCS or DVB-T standards.

BBHEADER Bits Re-Use

Among the 10 bytes of BBFRAME headers, at least three (SYNC and SYNCD) are not relevant for continuous Generic Streams (see Figure 4.6). Indeed, their use has been defined in the DVB-S2 standard for the sole purpose of allowing native transport of fixed-length PDUs over packetized Generic Streams. Their re-definition and use in the context of continuous Generic Streams might prove useful, and pave the way for further optimizations of future versions of the GSE protocol. Possible uses include: allowing further flow organization, stamping BBFRAMEs for e.g. Operation and Management (OAM) purposes or adapting MODCOD selection based on network layer QoS signaling.

Cross-Layer Enhancement of GSE's Error Control Techniques

It was shown in Chapter 3 that DVB-S2's enhanced FEC has lowered the ratio of undetectable to detectable errors to 10^{-8} in new generation satellites, making an undetected error event after FEC decoding extremely rare. For this reason, GSE could also benefit from the cross-layer mechanisms suggested here for DVB-S.

6.2.2 Future Directions for HERACLES

We believe that the HERACLES framework constitutes a promising research topic, given the excellent results obtained up to now. In order to come up with a working framework, a series of

particular choices were done during its development such as e.g. the use of cross-correlations or the definition of PSR as we defined it in Section 5.2.2. It is of course possible to reassess the relevance of these choices and try different alternatives to explore the possibilities offered by the mechanism.

There is with no doubt material for further improvements in this framework, and we basically lacked time to test more HERACLES configurations.

Extending the Header Redundancy Concept

A natural direction to examine for further improvement in HERACLES is the general use of *redundancy*, regardless its form and location in the flow. Indeed, redundancy is not only limited to the existence of STATIC header fields in a packetized flow, but can appear under many different ways in a message, both within packet headers and data as well.

For instance, one could imagine the existence of a function able to use the knowledge of the way the INFERRED fields change in headers within the flow, in order to complement the correction capabilities already brought by the STATIC part of the header. A practical example of this is given by Length fields: since SP recovery allows excellent flow delineation, any corrupted Length field can be recovered by counting the symbols separating two successive SP detections. For other INFERRED fields such as counters or flags, advanced studies on their variability exist for several known protocol stacks, since they are the basis over which general header compression schemes are built on.

In the general case, redundancy appears within the transmitted data as well, provided that it complies with the syntax rules of well identified languages (English, HTML, etc) or applications. Take the English language for example, whose redundancy patterns and structures have been widely studied from the beginning of communications theory. Video flows or HTML pages have also redundancy patterns in the forms of information tables or tags that, if properly identified and used, have the potential to contribute to the overall effort of future HERACLES-like mechanisms. In this regard, DUDE-like concepts for identifying structural data patterns in information flows are certainly a good starting point for further reflection.

Better FEC/HERACLES Integration

Section 5.5 discussed how placing the HERACLES block before the FEC decoder allowed for the latter to perform better. Future work could explore the possibilities of enhancing this HERACLES/FEC synergy e.g. by defining a joint decoding stage. Indeed, it seems interesting to analyze to which extent HERACLES can directly influence the decoding algorithm of the FEC subsystem, instead of just being a preliminary stage. Reciprocally, we have identified potential enhancements in the detection process of the HERACLES block with the use of FEC information: from this standpoint, the definition of a combined stage makes sense.

Explore Further HERACLES Integration Possibilities

The framework defined in Chapter 5 is general enough to serve as a basis for integrating HERACLES at different levels of the protocol stack. Section 5.5 focalized on a particular application case with a strong potential for system enhancement, but much work is still to be done in order to identify the best configurations for HERACLES use in a complete system, and especially, the best HERACLES configuration for every particular context. A thorough exploration of the different possibilities offered by the mechanism has the potential to redefine global error control strategies, with the possibility of achieving more balanced and effective policies for the sake of overall optimization.

Enriching HERACLES through Plug-ins

In addition to its core mechanisms, several "plug-ins" could be imagined to add new functionalities to HERACLES and enhance its overall performances. For instance, ETSW scans could be coupled with integrity checks (e.g. CRCs or hash mechanisms) calculated from the desired SP. One could then imagine a trial-and-error series of bit flips performed on selected sequences, in order to attempt reconstitution of the desired information. In another implementation, small error-correcting codes could be used to achieve such reconstitution locally.

Appendix A

Efficient IP over Second Generation Satellites (EIoSS)

A.1 Foreword

This Appendix introduces EIoSS, a series of experimental IP-centric ideas for encapsulating IP datagrams *directly* over a continuous Generic Stream [19]. It relies on an ambitious cross-layer approach to the problem making use of tunneling techniques, BBHEADER fields re-definition and the main thoughts on error control of Chapter 3. This unpublished framework was developed at the early stages of this PhD. work, and has been superseded by the posterior joint definition and standardization of GSE.

Undoubtedly, much work would have been needed to refine EIoSS. We are aware that the all-IP approach upon which it is built presents shortcomings from a networking standpoint (particularly regarding address resolution and security issues), some of which have been discussed within the IPDVB WG of the IETF [153]. It is included in this dissertation for informative purposes only, hoping that some its concepts may stimulate future thoughts on innovative IP mappings over any kind of generic stream.

A.2 Description of the Technique

A.2.1 Network Scenarios

We consider a network scenario with a transparent (non-regenerative) satellite providing multi-spot coverage over a large area, and for which every beam carries a multiplexed flow of BBFRAMEs with different MODCODs. Data gathered at IP-enabled Gateways is relayed via the satellite link to a series of satellite receivers connected in a meshed configuration. Some receivers can act as entry routers of IP sub-networks, such as e.g. a LAN or the Internet.

A.2.2 The Principle

General case

Instead of being a multi-purpose and multi-protocol encapsulation, EIoSS defines an interface of IP over Generic Streams, by mapping IPv4 and IPv6 packets *exclusively* over a flow of BBFRAMEs, without any kind of CRCs or encapsulation headers. For this to be possible, the gateway must process the flows at IP-level and then it must basically integrate the functionalities of an IP router.

Non-IP traffic support

Many real world implementations of commercial encapsulation devices perform IPv4/IPv6 filtering directly upon SNDU reception []. Therefore, since EIoSS is designed to carry only IPv4 and IPv6 data, it does not need a Type field to identify the payload carried. From time to time, however, non-IP traffic will have to be carried over the network, if we suppose that a Gateway (that acts basically as a router) is also capable of aggregating other kind of network layer packets. When needed, available techniques defined by the IETF can allow carrying packets of other network-layer protocols such as Ethernet or MPLS, by encapsulating them at the Gateway into IP datagrams authorized to travel along the satellite link. This technique is sometimes known as tunneling, and allows also carrying encrypted IP packets, such as IPsec in tunnel mode as described in RFC 2709 [154].

IP header compression support

As for IP header compression support, a Gateway having router functionalities will never have to *forward* IP header-compressed packets. In the case the Gateway receives IP header compressed datagrams, and since IP header compression is a hop-to-hop process, the Gateway (acting as a node) will always be able to reconstruct the compressed packets prior to satellite transmission. Of course, a general IP compression scheme can be applied by the Gateway to the overall traffic sent to the satellite, a particular issue that will be discussed later.

The consequences of this IP-centric parsing technique, as well as its limits and design implications will be studied in the last part of this Appendix. Through the following lines, we explain how the classical functions typically ensured by adaptation layers such as error control, fragmentation or addressing can be delegated either to the link or to the IP layers. The basic condition for this is allowing cross-layer techniques to be used in the overall process.

A.2.3 Processing at the Gateway

"Foo over IP" encapsulation

IP and non-IP flows are aggregated at the transmission Gateway. As a first step, non-IP traffic is encapsulated into IP datagrams using one of the numerous "foo over IP" techniques available. A strong design assumption is therefore that the non-IP vs. IP traffic ratio to be dealt with is small, and will keep shrinking. Otherwise the multiplication of IP headers used for encapsulation would decrease dramatically the efficiency of the system. Note that the support or not for carrying IP header compressed datagrams is not an issue, as explained in the previous section.

Addressing issues

IP traffic is separated into a series of logical channels according to IP routing criteria derived from the IP headers: the Gateway behaves as a router that has one or few broadband entries, but as many exits as logical channels. Logical channels are managed statically or dynamically by a Network Control Center (NCC), and could be allocated e.g. to an Internet Services Provider (ISP), a particular geographical region, an Autonomous System or a group of users sharing some similar routing characteristics. DVB-S2 allows natively the definition of $2^8 = 256$ of them, specified by the ISI field (8 bits) of every 10-byte BBHEADER. If desired, and since the UPL field is redundant for continuous GS, its 2 bytes could be used e.g. to provide an extension allowing for further discrimination of BBFRAMEs in a single logical channel.

QoS selection and MODCOD allocation

Next, QoS requirements for the IP datagrams are analyzed, and a scheduler distributes them accordingly into several buffers (as many as MODCODs currently in use in the system) independently for every logical channel. The fundamental fact that IP is a connectionless protocol and that EIoSS does not introduce any artificial boundary between IP and the scheduler allows the latter one to flexibly move, duplicate, delay or even drop datagrams for the sake of overall optimization. Knowing that integrity control, reordering and reassembly is ensured at the upper layers, the scheduler has practically no constraints for properly ensuring its task. Although the precise definition of the scheduling and MODCOD allocation is out of the scope of this framework, at the end of this process several IP flows are ready to be sliced and packed over BBFRAMEs, whose size varies for every MODCOD.

Link-level fragmentation: Threads and Pearls

We consider a flow of BBFRAMEs labeled $0, 1, ..., N$ sent over a logical channel. In the general case where the end of BBFRAME i does not correspond to the end of an IP packet d, link-level fragmentation takes place and d is fragmented without making use of a CRC, following the considerations of Chapter 3.

For the sake of clarity, we only consider 2 fragments d_1 and d_2. Fragment d_1 can be sent using the available bytes of BBFRAME i, and d_2 has to be sent *at the beginning* of BBFRAME j, where $j \geq (i+1)$. A *Payload Pointer (PP)* set in BBFRAME j will specify how many of its bytes were used for sending d_2. The remaining bytes of BBFRAME j can be used to send other datagrams (and possibly a datagram fragment) and so on, a procedure classically known as packing. However, if there is no available data to send, a stuffing procedure (padding) fills the remaining bytes of BBFRAME j with a pre-defined sequence (e.g. zeros). This function is coupled with a timer that ensures a reasonable "latency vs. optimal filling" trade-off for every BBFRAME.

In the case $j = (i+1)$, fragmentation is *consecutive*, similar to what is classically done in the stream-based MPEG2. However, particular scheduling policies may also require $j > (i+1)$: in this case, *non-consecutive fragmentation* of the datagrams occurs over the flow of BBFRAMEs. In order to support this enhanced flexibility, we introduce *Threads* and *Pearls*. A *Thread* is a logical group of BBFRAMEs starting with a null *PP*; the BBFRAME carrying this null *PP* is labeled *Pearl* 0. When a BBFRAME carries a datagram fragment whose previous fragment was carried by *Pearl* p of a *Thread* t, it becomes *Pearl* $(p+1)$ of *Thread* t. Finally, a *Thread* is terminated when one of its *Pearls* experiences padding or uses its last bytes without inducing fragmentation on a datagram. The final *Pearl* of every *Thread* is tagged with a *Stop Thread* bit. There can be several active *Threads* in the system, but a BBFRAME is always a unique *Pearl* of a unique *Thread*.

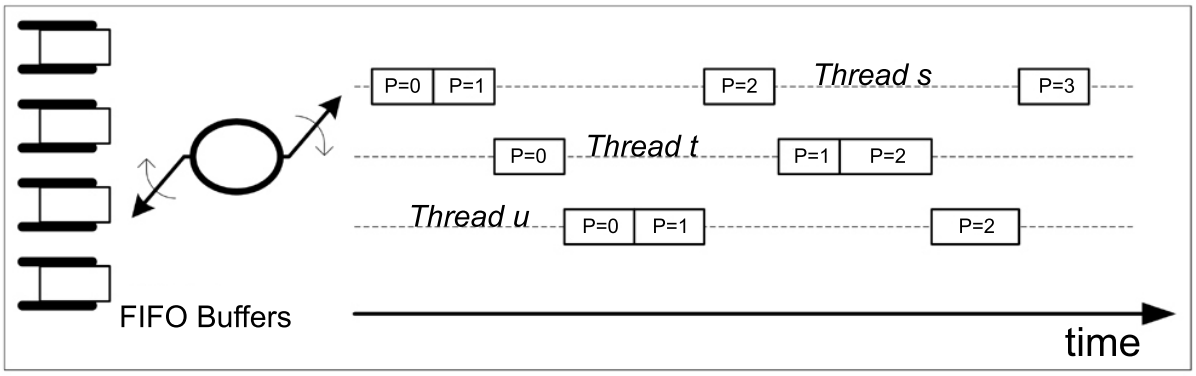


Figure A.1: Representation of *Pearls* and *Threads* in a simple system with only 3 *Threads* (note that BBFRAMEs do not necessarily have the same MODCODs, and therefore, the same sizes). FIFO buffers could be differentiated e.g. by QoS considerations and/or MODCOD.

This mechanism ensures that even in the case a datagram was fragmented and scattered over non-consecutive BBFRAMEs of the flow, its pieces can be put together at the receiver. Instead of defining an encapsulation header for this mechanism, we propose to indicate the *PP*, *Pearl*, *Thread* and *Stop Thread* for every BBFRAME using some of the available bits in every BBHEADER as shown in Figure A.3).

Finally, every BBFRAME is coded and modulated according to its initial MODCOD requirements, prior to be sent in a time-division multiplex.

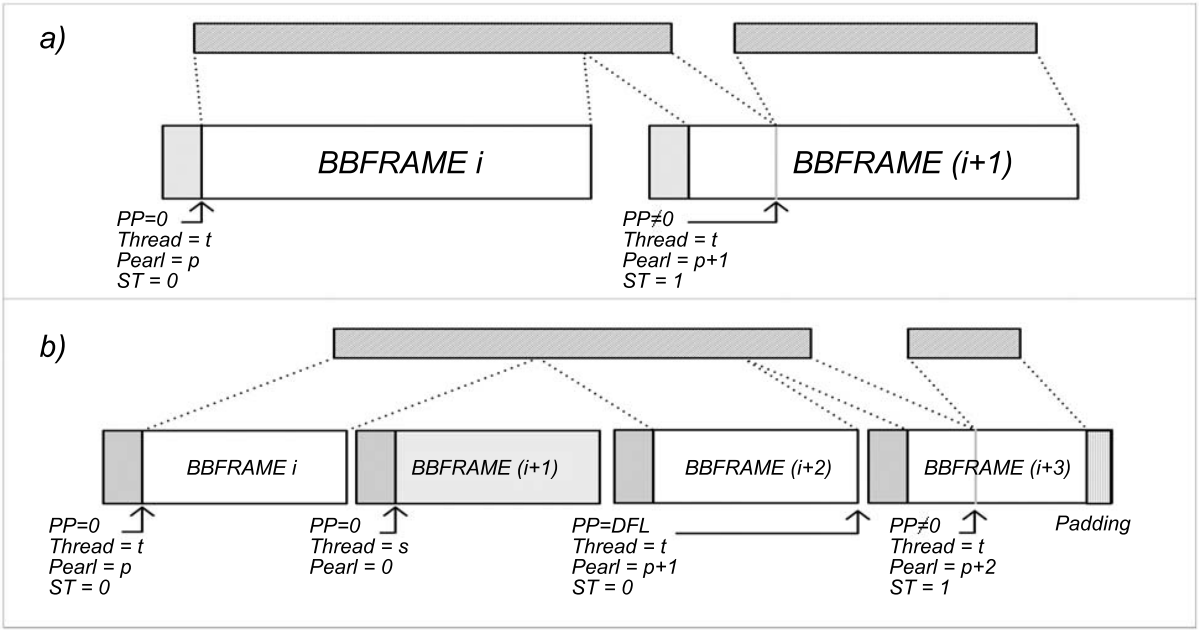


Figure A.2: Consecutive (a) and non-consecutive fragmentations (b). BBFRAMEs i, $(i+2)$ and $(i+3)$ are *Pearls* belonging to *Thread* t.

Figure A.3: Proposed modification of the SYNC and SYNCD fields in order to specify the *Thread*, *Pearl*, *Stop Thread*, and *PP* values. This allocation allows for 2^5 simultaneous *Threads* and 2^5 *Pearls* per *Thread*. As for *PP*, 13 bits are enough to be able to point to any position of the longest BBFRAME.

A.2.4 Processing at the Receiver

A receiver is a device able to tune a multiplex, and to listen to one or more logical channels from this multiplex. It can be a router or an end host.

BBFRAME demodulation and decoding

Every receiver demodulates the frames transmitted with a less efficient MODCOD than its own maximum supported MODCOD, and makes a first filtering based on the ISI field. The FEC decoder, configured for error correction and detection, is configured to discard (or tag) the BBFRAMEs it considers erroneous after FEC decoding, following the considerations of Chapter 3. Given the fact that this information is extremely reliable, the data passed to the unit in charge of extracting the IP packets out of the BBFRAMEs can be considered error-free.

Packet reassembly and network-level filtering

Upon BBFRAME reception, all the receivers start "threading" *Pearls* in order, separately for each *Thread*. For this, every receiver analyzes every BBHEADER, and applies the following 3 rules:

1. When a *Pearl* 0 belonging to a *Thread* t arrives, a new buffer and a timer associated with t are created, and all the bytes of *Pearl* 0 are placed in the buffer. Joint use of the version, length and destination fields of the IP headers allows every receiver to know exactly how many full IP packets are contained in the buffer, and who they are addressed to. Upon extraction and filtering of the existing full IP packets a few bytes remain in the buffer, belonging to an IP datagram awaiting for its remaining bytes in the following *Pearls*.

2. When a *Pearl* $p > 0$ arrives, its bytes are first appended to the remaining bytes in the buffer, left by *Pearl* $(p-1)$. Then, the value indicated by *PP* is cross-checked with the number of remaining bytes in the buffer, since their sum must match the LENGTH field of the first complete datagram in the buffer. Like for *Pearl* 0, full IP packets are finally extracted. The few bytes possibly remaining in the buffer await the remaining bytes in the following *Pearls* and the timer is reset.

3. Finally, if a *Stop Thread* arrives or the timer expires, the buffer is analyzed and destroyed.

The precise numbering of every BBFRAME as a particular *Pearl* of a single *Thread* allows every receiver to operate very accurate delineation, framing and network-level filtering for every datagram.

Non-IP traffic will only be transformed back into its original shape and delivered to the corresponding upper layer protocols at the end host, and thus will be routed upon reception as a normal IP datagram. Note that current solutions rely mostly on link-layer addressing (e.g. based on MAC address), since the multi-purpose nature of classical encapsulations makes difficult the access to the network-level addressing information contained in heterogeneous PDUs under different formats and positions. Link-layer addressing has the drawback of requiring the use of tables updated and broadcasted on a regular basis, incurring in additional mechanisms definition and bandwidth use.

Given the probable fact that under DVB-S2 most of the subscribers will be small or medium size terminals [81], the complexity of this operation should remain low in order for the terminals to remain affordable and the satellite offer to be competitive. A simple complexity evaluation can be performed as follows. A single TDM stream under DVB-S2 will be transmitted at a maximum rate of about 40 MBaud, meaning around 120 Mbit/s of raw data assuming a 3-ary modulation such as 8-PSK. If the resource is equally shared among the 256 ISI channels carrying purely unicast traffic, every channel will carry a share of the traffic roughly equivalent to 500 kbit/s (raw data). Listening and analyzing data at these speeds is well within the reach of any off-the shelf PC today.

A.3 Analysis of the EIoSS Solution

A.3.1 Advantages

Simple

EIoSS does not make use of any ad-hoc features such as error checks, continuity or framing control, classically duplicated in the encapsulation layers: it relies for this on the robustness of the FEC below and on the proven simplicity of IP for ensuring proper delivery of the datagrams. The only addition of EIoSS to DVB-S2 are *Threads* and *Pearls*: The goal is to introduce a clear logical organization of the flow in order for the receivers to correctly untangle the plait of BBFRAMEs they listen to, keeping in mind that non-consecutive fragmentation allows for optimal scheduler flexibility.

IP-centric

Given the explosive growth of the Internet over the past years, the IP vs. non-IP traffic ratio is likely to keep growing in digital communication networks. Adapting satellites once for all to support IP traffic seems therefore wiser than constantly adapting IP into formats that were initially designed for carrying different data such as e.g. broadcast contents (MPEG2). MPEG2 Transport Streams are maintained in DVB-S2 for distributing e.g. video and audio contents, but convergence of data services towards IP and better integration of IP-based network is likely to benefit from the use of an IP over GS approach such as EIoSS.

Flexible

The absence of systematic headers as the ones introduced in classical encapsulation layers is one of the major advantages of EIoSS. However, BBFRAMEs are generally long and even relatively bad encapsulations can perform well under the "overhead efficiency" criterion. The real advantage of the solution is its IP-based approach for flexibly dealing with functions that are usually replicated at the encapsulation layers while being robust, simple in principle and allowing a better interconnection of IP networks. In particular, the fact that the scheduler deals directly with IP datagrams or fragments allows it to cut, drop, move, delay or even repeat them, since proper delivery and reassembly will be directly ensured by the upper layers. The connectionless approach of IP is with no doubt the most suited asset for a flexible ACM operation.

Robust to losses and errors

DVB-S2 specifies the more robust and reliable physical layer available today for satellites, and the small 0.4 to 0.8 dB that separates its FEC from the Shannon are an excellent indicator of how reliable the transmission can be considered at acceptable data rates. However, changes in MODCOD or other scenarios might cause temporarily some BBFRAMEs not to be received. Nevertheless,

following the loss of a BBFRAME, a receiver can recover from the next one by just ignoring the first bytes indicated by its *Payload Pointer*. In addition, the logical division of the flow in *Threads* allows for early detection of potential errors and discarding of erroneous IP datagrams.

Efficient

EIoSS defines a method that does not introduce any overhead on a per-PDU basis. Of course, IP headers will be used for encapsulation of non-IP traffic and overhead might be generated locally. However, this is to be compared numerically to what would have resulted if every single datagram had to be systematically packed with an encapsulation layer and a trailer.

A.3.2 Drawbacks

Required Standard Revision

The major drawback with EIoSS is that it requires the re-definition and use of at least 2 BBHEADER fields, namely SYNC and SYNCD (possibly three, with UPL). Although non-relevant or redundant for continuous Generic Streams, their use in the EIoSS context would constitute a violation of the DVB-S2 standard...

Driver Complexity increase

Since EIoSS concentrates almost all the data analysis at the IP layer in both sides of transmission, processing linked to filtering might be increased in the network drivers. However, this is not likely to drive terminal complexity and prices up, since reassembly and decapsulation are considerably reduced. The real complexity increase is bounded, and can be dealt with using available hardware (IP routers mainly) and technology.

Limits of the all-IP approach

Several studies were conducted in the past in the aim of using a satellite as an IP router, and various interesting issues raised by this approach were analyzed. EIoSS is by no means such kind of solution, but some limitations of the IP-dedicated scheme studied in these analyses may have an impact on EIoSS. For instance, ARP requests over a LAN, which rely on L2-broadcasted frames over an Ethernet are good examples of such a problem, although cases of LANs integrating a satellite link are rather rare. Since EIoSS relies on the existence of a network level address to deliver data, it seems impossible for EIoSS to deliver a frame that does not know its IP address.

However, most of these problems can be addressed by IP-based means, keeping in mind the goal of improving the integration of IP services in a single global infrastructure. In the case of the ARP example, the creation of a dedicated L2/L3 address resolution server for the network (single or

distributed) whose L3 address is known can be a solution [155], just like the DNS works today. If such a system existed and kept track of the L2/L3 address parsing, any machines using the satellite link could retrieve the IP address of any other machine of the network using IP-only queries and replies, instead of using an ARP broadcast.

A.3.3 Natural Extensions

IP payload compression

Since EIoSS is an all-IP solution and the bandwidth and power resources of the satellite are scarce, introducing a good IP compression scheme adapted to EIoSS is an interesting issue. Long packets could benefit from payload compression techniques, as described in RFC 2393 [156].

IP header compression

For small packets (say, under 100 bytes), IP header compression may be an interesting solution. Many available schemes exist today such as the ones mentioned in Section 5.1. Integrating smoothly these schemes into EIoSS seems possible since most of them have defined point-to-multipoint profiles. However, since EIoSS relies on the continuity of uncompressed IPv4/IPv6 packets inside a BBFRAME for framing at the receiver, a short preamble must be introduced to wrap compressed datagrams. This will ensure proper framing information to be retrieved upon BBFRAME reception and "threading". We estimate this overhead to be around 2 bytes per IP header compressed packet distributed as follows:

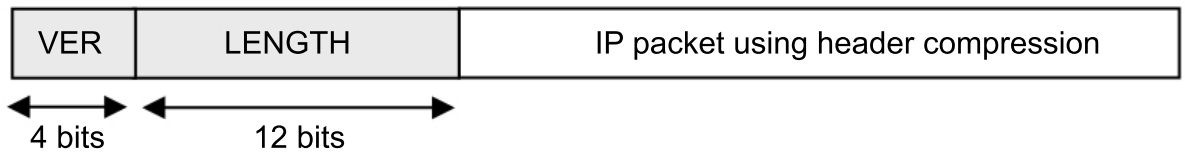


Figure A.4: Possible preamble for an IP header compressed datagram under EIoSS

The VER field has to be set to any value different from 4 or 6, so that upon insertion in the BBFRAME flow it cannot be mistakenly interpreted as the start of a normal IP header.

The LENGTH field is given to provide framing information to the receiver, who otherwise has no means to locate the start of the following IP packet in the BBFRAME flow. Indeed this information (as well as the IP address) cannot be found inside the compressed header without a priori knowledge of the compression parameters and context.

Extension for native support of other protocols

EIoSS is IP-centric since the convergence towards an all-IP world seems clear. However, in the case another predominant protocol emerged in the future years (e.g. Ethernet or MPLS), the framework of EIoSS could be easily extended in order to integrate it without any modification, provided that the gateway and the receivers support it natively.

Indeed, a smooth integration of a new flavor of PDUs into EIoSS only requires a clear indication of length and a way to differentiate them from IP datagrams. For this, and in order to stick to the IP format for the VERSION field, a 4-bit preamble could be used. As for packet delineation, if the protocol header does not contain any LENGTH field it can be explicitly added, just as suggested for the support of IP header compression. In any case, increased overhead should remain below 3 bytes.

A.4 Conclusion

EIoSS is an IP-centric set of proposals for the transmission of network layer packets over the forward link of DVB-S2 satellites. Instead of constantly adapting IP to satellite sub-networks not designed for IP at the detriment of efficiency and flexibility, EIoSS relies on the use of cross-layer techniques for mapping IP datagrams *directly* over the bearers of the new standard. The connectionless essence of IP optimizes the overall efficiency and offers an increased flexibility for the use of the ACM techniques. EIoSS takes advantage of all the main improvements of the new standard compared to DVB-S: the advanced error control provided by the improved FEC, the increased average size of DVB-S2 bearers and the possibility to flexibly manage the data to be transported for the sake of overall optimization. Up to now, classical link/physical layers on satellite links did not allow such kind of approaches, due to the stream-based nature of their lower layers (essentially suited for video and audio broadcast), the small size of their bearers and the impossibility to adapt their transmission parameters to the state of the network. Finally, in order to use at best the scarce power and bandwidth resources available on-board, EIoSS supports very well IP compression techniques — both for headers or payload — with the introduction of a two-byte preamble per packet. The concept of EIoSS could be easily extended to natively carry packets from other protocols, as long as they represent an important part of the traffic for the satellite link.

Bibliography

[1] ETSI. Digital Video Broadcasting (DVB); Modulation and coding for DBS satellites systems at 11/12 GHz (DVB-S), EN 301 421.

[2] ETSI. Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2), EN 302 307, 2004.

[3] F. Arnal. *Optimisation de la fiabilité pour des communications multipoints par satellite géostationnaire*. PhD thesis, ENST Paris, December 2004.

[4] A. Bolea Alamañac. *Conception et mise en oeuvre de méthodes de compensation des effets du canal de propagation pour optimiser les ressources radio*. PhD thesis, Supaero, Toulouse, December 2004.

[5] H. Nyquist. Certain factors affecting telegraph speed. *The Bell System Technical Journal*, 3:324, 1924.

[6] R.V. Hartley. Transmission of information. *The Bell System Technical Journal*, 7:535, 1928.

[7] C.E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[8] ETSI. Specifications for Data Broadcasting, EN 301 192.

[9] G. Fairhurst and B. Collini-Nocker. Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS). RFC 4326 (Proposed Standard), December 2005.

[10] D.A. Huffman. Dependency-Aware Unequal Erasure Protection Codes. In *Proceedings of the IRE*, volume 40, pages 1098–1101, September 1952.

[11] J. Ziv and A. Lempel. Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536, September 1978.

[12] T.A. Welch. A Technique for High Performance Data Compression. *IEEE Computer*, 17:8–19, June 1984.

[13] G. Buch, F. Burket, J. Hagenauer and B. Kukla. To compress or not to compress? In *Proceedings of the IEEE Global Telecommunications Conference*, 1996.

[14] J. Hagenauer. Source-Controlled Channel Decoding. *IEEE Transactions in Communications*, 43(9):2449–2457, September 1995.

[15] F. Hekland, G.E. Øien and T. A. Ramstad. Using 2:1 Shannon Mapping for Joint Source-Channel Coding. In *Proceedings of DCC*, pages 223–232. IEEE Computer Society, 2005.

[16] X. Cai and J. W. Modestino. Bandwidth Expansion Shannon Mapping for Analog Error-Control Coding. In *Proceedings of the* 40^{th} *Annual Conference on Information Sciences and Systems*, pages 407–412, 2006.

[17] J. Cantillo and J. Lacan. CRC-32 Performance Assessment for DVB-S Links. Technical report, Thales Alenia Space, May 2005.

[18] J. Cantillo and J. Lacan. Direct IP Over Generic Streams. Technical report, Thales Alenia Space, May 2005.

[19] J. Cantillo and J. Lacan. An Access Scheme for Providing Efficient IP Services over DVB-S2. Technical report, Thales Alenia Space, May 2006.

[20] J. Cantillo and J. Lacan. Robust Synchronization and Header Recovery in Upper Layers Using Header Redundancy. Technical report, Thales Alenia Space, February 2007.

[21] J. Cantillo and J. Lacan. A Design Rationale for Providing IP Services Over DVB-S2 Links. IETF draft, draft-cantillo-ipdvb-s2encaps-04.txt, expired, December 2007.

[22] J. Cantillo, J. Lacan and I. Buret. A CRC Usefulness Assessment for Adaptation Layers in Satellite Systems. In *Proceddings of AIAA's* 24^{th} *International Communications Satellite Systems Conference, San Diego, California*, June 2006.

[23] J. Cantillo, J. Lacan, I. Buret and F. Arnal. Design Issues for the Generic Stream Encapsulation (GSE) of IP Datagrams over DVB-S2. In *Proceedings of the* 4^{th} *International Workshop on Satellite and Space Communications (IWSSC-07)*, 2007.

[24] J. Cantillo, J. Lacan and I. Buret. Cross-layer enhancement of error control techniques for adaptation layers of DVB satellites. *International Journal of Satellite Communications and Networking*, 24:579–590, 2006.

[25] J. Cantillo, B. Collini-Nocker, U. De Bie, O. Del Rio, G.Fairhurst, A. Jahn and R. Rinaldo. GSE: A Flexible, yet Efficient, Encapsulation for IP over DVB-S2 Continuous Generic Streams. *International Journal of Satellite Communications and Networking*, To appear, 2008.

[26] J. Cantillo, J. Lacan, I. Buret and F. Arnal. Procédé et module de correction d'erreurs de transmission dans un flux de données, système de communication comprenant ledit module, December 2007. French patent application FR0708623.

[27] J. Cantillo, J. Lacan, I. Buret and F. Arnal. Procédé et dispositif de délinéation d'un flux de données et système de communication comprenant ledit dispositif, 2008. Ongoing application.

[28] DVB. Generic Stream Encapsulation (GSE) Protocol. DVB BlueBook A116, draft ETSI TS 102 606, May 2007.

[29] ISO. Information Processing Systems - Open Systems Interconnection - Basic Reference Model, American National Standards Institute, ISO/IEC 7498-1, 1984.

[30] Wikipedia. http://www.wikipedia.org.

[31] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (Standard), October 1989. Updated by RFCs 1349, 4379.

[32] B. Carpenter. Architectural Principles of the Internet. RFC 1958 (Informational), June 1996. Updated by RFC 3439.

[33] R. Bush and D. Meyer. Some Internet Architectural Guidelines and Philosophy. RFC 3439 (Informational), December 2002.

[34] M. Allman, D. Glover, and L. Sanchez. Enhancing TCP Over Satellite Channels using Standard Mechanisms. RFC 2488 (Best Current Practice), January 1999.

[35] G. Maral and M. Bousquet. *Satellite Communications Systems - Systems, Techniques and Technology*. John Wiley & sons, 4^{th} edition, 2002.

[36] ETSI. Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM) services and architectures; Functional architecture for IP internetworking with BSM networks, TS 102 292, V1.1.1 (2004-02).

[37] A. Frank. *Cross-Layer Design Tutorial*. Norwegian University of Science and Technology, Dept. of Electronics and Telecommunications, Trondheim, Norway, November 2004. Published under Creative Commons License.

[38] V. Srivastava and M. Motani. Cross-Layer Design: a Survey and the Road Ahead. *IEEE Communications Magazine*, 43(12):112–119, 2005.

[39] S. Kota and G. Giambene. QoS for IP-Based Satellite Networks: Cross-Layer Design. In *Proceedings of the* 24^{th} *AIAA International Communications Satellite Systems Conference (ICSSC)*, June 2006.

[40] ETSI. Transmission System for Handheld Terminals (DVB-H), EN 302 304.

[41] ETSI. System Specifications for Satellite services to Handheld devices (SH) below 3 GHz (draft TS 102 585 V1.1.1).

[42] ETSI. Framing Structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3 GHz (draft EN 302 583 V1.1.1).

[43] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Proposed Standard), September 2001.

[44] G. Giambene and S. Kota. Cross-layer Protocol Optimization for Satellite Communications Networks: a Survey. *International Journal of Satellite Communications and Networking*, 24:323–341, 2006.

[45] P. Sarolahti, S. Floyd and M. Kojo. Transport-Layer Considerations for Explicit Cross-Layer Indications. IETF draft, draft-sarolahti-tsvwg-crosslayer-01.txt, expired, September 2007.

[46] D. Katabi, M. Handley and C. E. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. In *Proceedings of SIGCOMM*, pages 89–102. ACM, 2002.

[47] M. Rossi, R. Vicenzi and M. Zorzi. Accurate Analysis of TCP on Channels With Memory and Finite Round-Trip Delay. *IEEE Transactions on Wireless Communications*, 3(2):627–640, March 2004.

[48] P. Karn, C. Bormann, G. Fairhurst, D. Grossman, R. Ludwig, J. Mahdavi, G. Montenegro, J. Touch, and L. Wood. Advice for Internet Subnetwork Designers. RFC 3819 (Best Current Practice), July 2004.

[49] V. T. Raisinghani and S. Iyer. Cross-layer design optimizations in wireless protocol stacks. *Computer Communications*, 27(8):720–724, 2004.

[50] ISO. Information Technology; Generic Coding of Moving Pictures and Associated Audio Information Systems, ISO/IEC DIS 13818-1 , 2000.

[51] ETSI. Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems (DVB-RCS), EN 301 790.

[52] P. Chini, G. Giambene, D. Bartolini, M. Luglio and C. Rosetti. Dynamic Resource Allocation Based on a TCP-MAC Cross-Layer Approach for DVB-RCS Satellite Networks. *International Journal of Satellite Communications and Networking*, 24:367–385, 2006.

[53] C. Comaniciu D. Wang and U. Tureli. Cross-layer design for localization and MAC. In *Proceedings of the* 40^{th} *Annual Conference on Information Sciences and Systems*, pages 407–412, 2006.

[54] X. Lin, N.B. Shroff and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, 2006.

[55] B. Radunovic. *A Cross-Layer Design of Wireless Ad-Hoc Networks*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2005.

[56] S. Mérigeault and C. Lamy. Concepts for exchanging extra information between protocol layers transparently for the standard protocol stack. In *Proceedings of IEEE ICT'03, Tahiti, French Polynesia*, February 2003.

[57] G. Fairhurst, M. Berioli and G. Renker. Cross-Layer Control of Adaptive Coding and Modulation for Satellite Internet Multimedia. *International Journal of Satellite Communications and Networking*, 24:471–491, 2006.

[58] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

[59] S. Floyd, M. Allman, A. Jain, and P. Sarolahti. Quick-Start for TCP and IP. RFC 4782 (Experimental), January 2007.

[60] M. Allman P. Sarolahti and S. Floyd. Determining an Appropriate Sending Rate over an Underutilized Network Path. *Computer Networks*, 51(7):1815–1832, 2007.

[61] P. Sarolahti, J. Korhonen, L. Daniel and M. Kojo. Using Quick-Start to Improve TCP Performance with Vertical Hand-offs. In *Proceedings of the* 31^{st} *IEEE Conference on Local Computer Networks*, pages 897–904, 2006.

[62] H. Balakrishnan and R.H. Katz. Explicit Loss Notification and Wireless Web Performance. In *Proceedings of the IEEE Globecom Internet Mini-Conference, Sidney, Australia*, November 1998.

[63] R. Krishnan, M. Allman, C. Partridge, J. Sterbenz and W. Ivancic. Explicit Transport Error Notification (ETEN) for ErrorProne Wireless and Satellite Networks.

[64] D. Katabi. Specification for the Explicit Control Protocol (XCP). IETF draft, draft-falk-xcp-spec-00.txt, expired, October 2004.

[65] Explicit Congestion Protocol. http://kb.pert.geant2.net/PERTKB/ExplicitCongestionProtocol.

[66] L. Wu, F. Peng and V. C. M. Leung. Dynamic Congestion Control to Improve Performance of TCP Split-Connections over Satellite Links. In *Proceedings of the International Conference On Computer Communications and Networks (ICCCN 2004), October 11-13, 2004, Chicago, IL, USA*, pages 268–275. IEEE, 2004.

[67] W. Stanislaus, G. Fairhurst and J. Radzik. Cross-Layer Techniques for Flexible Transport Protocols (UDP-Lite and DCCP) over a Satellite Network. In *Proceedings of the International Workshop on Satellite and Space Communications (IWSSC-05)*, 2005.

[68] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson, and G. Fairhurst. The Lightweight User Datagram Protocol (UDP-Lite). RFC 3828 (Proposed Standard), July 2004.

[69] S. Ramachandran, G. Fairhurst, M. Luglio, C. Roseti and S. Provenzano. Network Layer Security: Design for a Cross-Layer Architecture. In *Proceedings of the International Workshop on Satellite and Space Communications (IWSSC-07)*, 2007.

[70] F. Arnal, L. Dairaine, J. Lacan and G. Maral. Cross-Layer Reliability Management for Multicast over Satellite. In *Proceedings of Computer Networks*, 2004.

[71] T. Connolly, P. Amer, and P. Conrad. An Extension to TCP : Partial Order Service. RFC 1693 (Experimental), November 1994.

[72] P.D. Amer, C. Chassot, T.J. Connolly, M. Diaz and P. Conrad. Partial-Order Transport Service for Multimedia and Other Applications. *IEEE/ACM Transactions on Networking*, 2(5):440–456, October 1994.

[73] A. Bouabdallah and J. Lacan. Dependency-Aware Unequal Erasure Protection Codes. In *Proceedings of the* 15^{th} *Packet Video Workshop, Hanghzou, China*, April 2006.

[74] G. Liebl, et al. An RTP Payload Format for Erasure-Resilient Transmission of Progressive Multimedia Streams. IETF draft, draft-ietf-avt-uxp-07.txt, expired, October 2004.

[75] F. Vieira, M.A. Vazquez Castro and G. Seco Granados. A Tunable-Fairness Cross-Layer Scheduler for DVB-S2. *International Journal of Satellite Communications and Networking*, 24:437–450, 2006.

[76] Q. Liu, X. Wang and G.B. Giannakis. A Cross-Layer Scheduling Algorithm with QoS Support in Wireless Networks. *IEEE Transactions on Vehicular Technology*, 55(3):839–847, 2006.

[77] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936, 4495.

[78] H. Schulzrinne and R. Hancock. GIST: General Internet Signaling Transport. IETF draft, draft-ietf-nsis-ntlp-14.txt, work in progress, July 2007.

[79] L. Castanet, A. Bolea and M. Bousquet. Interference and Fade Mitigation Techniques for Ka Band and Q/V Band Satellite Communications Systems. In *Proceedings of the International Workshop of COST, Action 280*, 2003.

[80] R. Rinaldo and R. de Gaudenzi. Capacity analysis and system optimization for the forward link of multi-beam satellite broadband systems exploiting adaptive coding and modulation. *International Journal of Satellite Communications and Networking*, 22:401–423, 2004.

[81] R. Rinaldo, M. A. Vazquez-Castro and A. Morello. DVB-S2 ACM modes for IP and MPEG unicast applications. *International Journal of Satellite Communications and Networking*, 22:367–399, 2004.

[82] A. Morello and U. Reimers. DVB-S2, the Second Generation Standard for Satellite Broadcasting and Unicasting. *International Journal of Satellite Communications and Networking*, 22:249–268, 2004.

[83] J. Wang, L. Li, S.H. Low and J.C. Doyle. Cross-layer optimization in tcp/ip networks. *IEEE/ACM Transactions on Networking*, 13(3):582–595, 2005.

[84] S. Shakkottai, T.S. Rappaport and P.C. Karlsson . Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10):74–80, 2003.

[85] N. Celandroni, F. Davoli, E. Ferro and A. Gotta. Networking with Multi-Service GEO Satellites: Cross-Layer Approaches for Bandwidth Allocation. *International Journal of Satellite Communications and Networking*, 24:387–403, 2006.

[86] G. Giambene and E. Zoli. Stability Analysis of an Adpative Packet Access Scheme for Mobile Communications Systems with High Propagation Delays. *International Journal of Satellite Communications and Networking*, 21:199–225, 2003.

[87] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), August 2002. Updated by RFC 4721.

[88] B. Aboba. Architectural Implications of Link Indications. RFC 4907 (Informational), June 2007.

[89] V. Kawadia and P.R. Kumar. A cautionary perspective on cross layer design. *IEEE Wireless Communications*, 12(1):3–11, February 2005.

[90] R. Callon. The Twelve Networking Truths. RFC 1925 (Informational), April 1996.

[91] D.D. Clark. Fault isolation and recovery. RFC 816, July 1982.

[92] The Cooperative Association for Internet Data Analysis (CAIDA). http://www.caida.org/home.

[93] S. Lin and D.J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1983.

[94] B. Friedrichs. *Kanalcodierung, Grundlagen und Anwendungen in modernen Kommunikationssystemen*. Springer Verlag, 1996.

[95] T. Kasami and S. Lin. On the Probability of Undetected Error for the Maximum Distance Separable Codes. *IEEE Transactions on Communications*, COM-32(9):998–1006, September 1984.

[96] S.K. Leung-Yan-Cheong and M. E. Hellman. Concerning a Bound on Undetected Error Probability. *IEEE Transactions on Information Theory*, pages 235–237, March 1976.

[97] R.T. Braden, D.A. Borman, and C. Partridge. Computing the Internet checksum. RFC 1071, September 1988. Updated by RFC 1141.

[98] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321 (Informational), April 1992.

[99] IT++ Website. http://itpp.sourceforge.net/current/installation.html.

[100] M.G. Kim and J.H. Lee. Undetected Error Probabilities of Binary Primitive BCH Codes for Both Error Correction and Detection. *IEEE Transactions on Communications*, 44(5):575–580, May 1996.

[101] A. Jahn. Procedure for Comparative Evaluation of IP/DVB-S2 Encapsulation Protocol over Generic Streams, December 2005. Personal Communication.

[102] DVB. Generic Stream Encapsulation (GSE) Protocol Implementation Guidelines, v4. Work in Progress, February 2008.

[103] ETSI. Digital Video Broadcasting (DVB); User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) TR 102 376, 2005.

[104] M. Eroz, F.-W. Sun and L.-N. Lee. DVB-S2 Low Density Parity Check Codes with Near Shannon Limit Performance. *International Journal of Satellite Communications and Networking*, 22:269–279, 2004.

[105] E. Casini, R. De Gaudenzi and A. Ginesi. DVB-S2 modem algorithms design and performance over typical satellite channels. *International Journal of Satellite Communications and Networking*, 22:281–318, 2004.

[106] F.-W. Sunz, Y. Jiangn and L.-N. Lee. Frame synchronization and pilot structure for second generation DVB via satellites. *International Journal of Satellite Communications and Networking*, 22:319–339, 2004.

[107] E. Chen, J. L. Koslov, V. Mignone and J. Santoru. DVB-S2 backward-compatible modes: a bridge between the present and the future. *International Journal of Satellite Communications and Networking*, 22:341–365, 2004.

[108] R. Gallager. Low Density Parity Check Codes. *IRE Transactions on Information Theory*, 1962.

[109] D.J.C. MacKay and R.M. Neal. Good codes based on very sparse matrices. In *Proceedings of the* 5^{th} *Cryptography and Coding IMA Conference*, pages 100–111, 1995.

[110] D.J.C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, March 1999.

[111] R. Rinaldo, A. Ginesi, R. De Gaudenzi, O. Del Rio, T. Flo, B. Rislow and H. P. Lexow. Advanced Physical and MAC Layer Techniques for DVB-based Interactive Satellite Terminals. In *Proceedings of the* 25^{th} *AIAA International Communications Satellite Systems Conference*, May 2007.

[112] M.-J. Montpetit, G. Fairhurst, H. Clausen, B. Collini-Nocker, and H. Linder. A Framework for Transmission of IP Datagrams over MPEG-2 Networks. RFC 4259 (Informational), November 2005.

[113] G. Fairhurst and M. Montpetit. Address Resolution Mechanisms for IP Datagrams over MPEG-2 Networks. RFC 4947 (Informational), July 2007.

[114] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), December 1998. Updated by RFC 5095.

[115] the IP over DVB (IPDVB) Charter Page. http://www.ietf.org/html.charters/ipdvb-charter.html.

[116] H. Cruickshank, S. Iyengar and P. Pillai. Security Requirements for the Unidirectional Lightweight Encapsulation (ULE) protocol. IETF IPDVB WG draft draft-ietf-ipdvb-sec-req, work in progress, 2007.

[117] O. Del Rio, A. Ginesi, R. Rinaldo, T. Flo and O. Weum. EDGE: Efficient DVB-S2 Generic Stream Encapsulation scheme, December 2005. Personal Communication.

[118] G. Fairhurst. An Encapsulation for Transmission of IP Datagrams over a DVB-S2 Generic Stream (GULE). IETF draft, draft-fairhurst-ipdvb-s2-gule-02.txt, expired, December 2005.

[119] G. Fairhurst and B. Collini-Nocker. Extension Formats for Unidirectional Link Encapsulation (ULE) and the Generic Stream Encapsulation (GSE). IETF IPDVB WG draft draft-ietf-ipdvb-ule-ext, work in progress, 2008.

[120] J. Lei, G. Seco Granados and M.A. Vazquez Castro. MPE/ULE-FEC vs GSE-FEC Efficiency Comparison of IP Datagram Transmission over DVB-S2. In *Proceedings of the* 25^{th} *AIAA International Communications Satellite Systems Conference*, May 2007.

[121] A. Mayer, B. Collini-Nocker, F. Vieira, J. Lei and M.A. Vazquez Castro. Analytical and Experimental IP Encapsulation Efficiency Comparison of GSE, MPE, and ULE over DVB-S2. In *Proceedings of the International Workshop on Satellite and Space Communications (IWSSC-07)*, 2007.

[122] ETSI. Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television (DVB-T), EN 300 744.

[123] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu and M. Weinberger. Universal Discrete Denoising: Known Channel. *IEEE Transactions on Information Theory*, 51, 2005.

[124] V. Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. RFC 1144 (Proposed Standard), February 1990.

[125] M. Degermark, B. Nordgren, and S. Pink. IP Header Compression. RFC 2507 (Proposed Standard), February 1999.

[126] M. Degermark, M. Engan, B. Nordgren and S. Pink. Low-loss TCP/IP Header Compression for Wireless Networks, 1996.

[127] S. Casner and V. Jacobson. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. RFC 2508 (Proposed Standard), February 1999.

[128] ROHC Charter Page. http://www.ietf.org/html.charters/rohc-charter.html.

[129] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed. RFC 3095 (Proposed Standard), July 2001. Updated by RFCs 3759, 4815.

[130] M. West and S. McCann. TCP/IP Field Behavior. RFC 4413 (Informational), March 2006.

[131] R.N. McDonough and A.D. Whalen. *Detection of Signals in Noise, 2^{nd} ed.* AT&T Bell Laboratories and Academic Press, 1971.

[132] A.L. Philippot. *Forme(s) d'ondes et accès adaptés au transport de paquets de taille variable pour une liaison satellite*. PhD thesis, ENST Paris, October 2005.

[133] A.L. Philippot, M.L. Boucheret, K. Leconte, C. Morlet, C. Bazile and J. Yu. Variable Size Packets (VSP) for transport of IP datagrams over satellite links: Physical and access layer optimization. *Space Communications*, 20(3-4):147–154, October 2006.

[134] K. Leconte and C. Morlet. Optimization of Satellite Access Lower Layers for the Transport of IP Datagrams. *IEEE Journal on Selected Areas in Communications*, 22(3):529–537, April 2004.

[135] V. A. Rodin and E. M. Semyonov. Rademacher series in symmetric spaces. *Analysis Mathematica*, 1(3):207–222, September 1975.

[136] S.J. Montgomery-Smith. The Distribution of Rademacher Sums.

[137] C. Berrou, A. Glavieux and P. Thitimajshima. Near Shannon limit Error-Correcting and Decoding: Turbo-code. *International Conference on Communications (ICC), IEEE*, May 1993. pp. 1064-1070.

[138] 3^{rd} Generation Partnership Project (3GPP) Technical Specification Group. Universal Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD), TS 25.212 v3.4.0. Also referred to as: European Telecommunications Standards Institute (ETSI) TS 125 212.

[139] J. G. Harrison. Implementation of a 3GPP Turbo Decoder on a Programmable DSP Core. In *Proceedings of the Communications Design Conference, San Jose, California*, October 2001.

[140] N. Letzepis and A. Grant. Bit Error Rate Estimation for Turbo Decoding. In *Proceedings of ISIT*, June-July 2003.

[141] J. Laster, J. Reed and W.H. Tranter. Bit Error Estimation Using Probability Density Function Estimators. *IEEE Transactions on Vehicular Technology*, 51(1):260–267, January 2003.

[142] D.R. Pauluzzi and N.C. Beaulieu. A comparison of SNR estimation techniques for the AWGN channel. *IEEE Transactions on Communications*, 48:1681–1691, October 2000.

[143] H. Hosseini and B. Rohani. Objective Characterization of Voice Service Quality in Wideband CDMA. In *Proceedings of the Vehicular Technology Conference*, 2001.

[144] C.E. Shannon. Prediction and Entropy of Printed English. *The Bell System Technical Journal*, 30:50–54, 1951.

[145] J. Cantillo, J. Lacan and M.L. Boucheret. HERACLES: Header Redundancy Assisted Cross-Layered Error Suppression, 2008. Submitted.

[146] S.A. Khayam, M.U. Ilyas, K. Pörsch, S. Karande and H. Radha. A Statistical Receiver-based Approach for Improved Throughput of Multimedia Communications over Wireless LANs. In *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2005.

[147] S.A. Khayam, S. Karande, M.U. Ilyas and H. Radha. Improving Wireless Multimedia Quality using Header Detection with Priors. In *Proceedings of the IEEE International Conference on Communications (ICC)*, June 2006.

[148] R.A. Bourne and C.I. Phillips. Fast IP Packet Delineator. *Electronics Letters*, 37(25):1557–1559, 2001.

[149] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu and M. Weinberger. Discrete Universal Denoising With Error Correction Coding, December 2005. US patent No. 2005/0289433.

[150] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu and M. Weinberger. Discrete Universal Denoising With Reliability Information, December 2005. US patent No. 2005/0289406.

[151] Erlenborn and Mark. Method and apparatus for combined framing and packet delineation, May 2004. US patent No. 2004/0100992.

[152] F. Walls and T. Spieker. System and method for detection and recovery of false synchronization using packet header information, February 2004. US patent No. 2004/0030967.

[153] Open Discussion Board for IETF's IPDVB Working Group. http://www.erg.abdn.ac.uk/ipdvb/archive/0508/maillist.html.

[154] P. Srisuresh. Security Model with Tunnel-mode IPsec for NAT Domains. RFC 2709 (Informational), October 1999.

[155] S. Combes, S. Josset, P. Very, I. Buret and T. Zein. Network scenarios for the IP-Dedicated Satellite Access Scheme. In *Proceedings of The* 8^{th} *Ka Band Utilization Conference, Baveno, Italy*, September 2002.

[156] A. Shacham, R. Monsour, R. Pereira, and M. Thomas. IP Payload Compression Protocol (IPComp). RFC 2393 (Proposed Standard), December 1998. Obsoleted by RFC 3173.