Network Coding for Efficient Vertical Handovers

R. Bassoli

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

Institute for Communication Systems Faculty of Engineering and Physical Sciences University of Surrey Guildford, Surrey GU2 7XH, U.K.

September 2015

© R. Bassoli 2015
Abstract

In 2008, Institute of Electrical and Electronics Engineers (IEEE) published its standard IEEE 802.21 for media-independent handover services. The main scope of this work was to design a technology agnostic mobility platform to perform vertical handovers between heterogeneous networks. Regarding vertical handover procedures, a key issue to address is the control of packet loss, which is responsible for high handover latency and low communication quality. The solution proposed by the standard IEEE 802.21 guarantees reliability by exploiting Automatic Repeat Request (ARQ). However, the use of an acknowledgement service has been demonstrated not to be the best way to handle frame loss.

In this thesis, we propose a novel architecture and protocol to efficiently perform vertical handovers. This protocol is called Enhanced-Coded MIH (EC-MIH) and exploits Forward Error Correction (FEC) instead of ARQ. In fact, it performs built-in coding operations to handle erasures of MIH frames. Moreover, we designed a novel hybrid concatenated coding scheme called Hybrid Serial Concatenated Network Code (HSCNC), composed of the serial concatenation of a classical erasure code and systematic Random Linear Network Coding (RLNC). We show via theoretical analysis as well as MATLAB simulations that the concatenation approach can outperform RLNC alone in terms of decoding error probability.

Moreover, this work analyses the frame loss of Media-Independent Handover (MIH) protocol during vertical handovers via system level simulations. The proposed HSCNC design is then integrated into the new EC-MIH protocol and evaluated. We then discuss how the new protocol outperforms the legacy protocol in terms of throughput (at TCP layer, above MIH) and handover delay.

Key words: Vertical handovers, IEEE 802.21, media-independent handover protocol, random linear network coding, network error correction codes, burst-erasure codes, concatenated codes.

Email: r.bassoli@surrey.ac.uk
WWW: http://www.eps.surrey.ac.uk/
Acknowledgements

My foremost gratitude is to my supervisors Dr. Jonathan Rodriguez, Dr. Sciamak Vahid and Professor Rahim Tafazolli, for their suggestions, their support and their guidance. I learnt invaluable knowledge from their experience.

Moreover, I would like to thank the FP7 Marie Curie Greenet project, the Instituto de Telecomunicações and the University of Surrey for the financial support and for the possibility to pursue my doctoral degree.

I want to thank my colleagues of Greenet project and of 4TELL group for their help, their support and the time we spent together in labs and in meetings. I will always keep with me those unforgettable moments. Especially, I am grateful to Hugo Marques for his useful comments during our collaborations.

Next, I want to thank Victor Sucasas, my colleague, my flatmate and my best friend, who shared with me all the experiences and the moments of this doctoral program and of the last four years.

Last but not least, I would dedicate this thesis to my family and my parents. Without their constant support it would have been impossible to achieve this special goal of my life.
Contents

1 Introduction .................................................. 1
  1.1 Overview on Vertical Handovers .......................... 1
      1.1.1 Error-Correcting Codes for Burst Erasure Channels .... 4
      1.1.2 Network Coding ...................................... 5
  1.2 Research Challenges ..................................... 9
      1.2.1 Challenges in Vertical Handovers ....................... 9
      1.2.2 Challenges in Network Error Correction Codes .......... 10
      1.2.3 Challenges in Burst Erasure Codes .................... 11
  1.3 Thesis Contribution ..................................... 12
      1.3.1 Forward Error Correction for Efficient Vertical Handovers .. 12
      1.3.2 Novel Hybrid Coding Scheme for Forward Error Correction .. 13
      1.3.3 Hybrid Coding for EC-MIH Protocol .................... 14
  1.4 Organisation of the Thesis .............................. 16

2 IEEE 802.21 Standard and Error-Correcting Codes for Burst Erasure Channels ......... 17
  2.1 Existing approaches to Vertical Handovers ................ 18
      2.1.1 IEEE 802.21: Media-Independent Handover Services .... 20
      Media-Independent Handover Protocol ....................... 23
  2.2 Burst Erasure Channel .................................... 25
      2.2.1 Gilbert’s Channel ................................... 27
  2.3 ‘Classical’ Error-Correcting Codes ....................... 28
      2.3.1 Reed-Solomon Codes ................................ 29
      2.3.2 Low-Density Parity-Check Codes .................... 31
Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3 Fountain Codes</td>
<td>34</td>
</tr>
<tr>
<td>Luby Transform Codes</td>
<td>35</td>
</tr>
<tr>
<td>Raptor Codes</td>
<td>37</td>
</tr>
<tr>
<td>2.4 Network Coding and Network Error-Correcting Codes</td>
<td>42</td>
</tr>
<tr>
<td>2.5 Summary</td>
<td>46</td>
</tr>
<tr>
<td>3 Hybrid Serial Concatenated Network Codes for Burst Erasure Channels</td>
<td>47</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>48</td>
</tr>
<tr>
<td>3.2 Luby Transform Codes and Random Linear Network Coding</td>
<td>53</td>
</tr>
<tr>
<td>3.3 Reed-Solomon Codes and Random Linear Network Coding</td>
<td>67</td>
</tr>
<tr>
<td>3.4 Low Density Parity Check Codes and Random Linear Network Coding</td>
<td>72</td>
</tr>
<tr>
<td>3.5 Raptor Codes and Random Linear Network Coding</td>
<td>76</td>
</tr>
<tr>
<td>3.5.1 Raptor Codes with Low Density Parity Check Codes</td>
<td>77</td>
</tr>
<tr>
<td>3.5.2 Raptor Codes with Reed-Solomon Codes</td>
<td>77</td>
</tr>
<tr>
<td>3.6 Summary and Conclusion</td>
<td>84</td>
</tr>
<tr>
<td>4 Hybrid Coding for Enhanced-Coded MIH (EC-MIH) Protocol</td>
<td>88</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>88</td>
</tr>
<tr>
<td>4.1.1 A Model for Vertical Handovers</td>
<td>90</td>
</tr>
<tr>
<td>4.2 General Architecture</td>
<td>95</td>
</tr>
<tr>
<td>4.3 Packet Structure</td>
<td>101</td>
</tr>
<tr>
<td>4.4 Overhead Analysis</td>
<td>103</td>
</tr>
<tr>
<td>4.5 Performance of the Protocol</td>
<td>104</td>
</tr>
<tr>
<td>4.5.1 Luby Transform Outer Codes</td>
<td>112</td>
</tr>
<tr>
<td>4.5.2 Reed-Solomon Outer Codes</td>
<td>116</td>
</tr>
<tr>
<td>4.5.3 Raptor Outer Codes</td>
<td>119</td>
</tr>
<tr>
<td>4.6 Summary and Conclusion</td>
<td>122</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>126</td>
</tr>
<tr>
<td>5.1 Main achievements</td>
<td>126</td>
</tr>
<tr>
<td>5.2 Future work</td>
<td>129</td>
</tr>
<tr>
<td>Bibliography</td>
<td>132</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 Taxonomy of vertical handover solutions ........................................... 2
1.2 Evolution of network coding theory .................................................... 7

2.1 IEEE 802.21 reference model ............................................................. 20
2.2 IEEE 802.21 architecture ................................................................. 22
2.3 State machines interactions of MIH protocol ........................................ 23
2.4 MIH general protocol frame format and its header .................................. 25
2.5 Gilbert’s model for burst error channels .............................................. 27
2.6 Folding of a RS code with parameter $\theta$ ........................................... 30
2.7 Simple example of an LDPC code ....................................................... 31
2.8 Simple example of superposition method to build a parity check matrix ... 33
2.9 Simple example of an LT code ............................................................ 35
2.10 Simulations to show how parameter $\delta$ affects the Robust Soliton distri-
    bution................................................................. 38
2.11 Simulations to show how parameter $c$ affects the Robust Soliton distri-
    bution................................................................. 39
2.12 Characterisation of the Robust Soliton distribution with $\delta = 0.1$ and
    $c = 0.1$. ........................................................................ 40
2.13 Characterisation of the Robust Soliton distribution with $\delta = 0.0001$ and
    $c = 0.06$. ........................................................................ 41
2.14 Comparison between different formulations of a linear code multicast ... 43

3.1 Point to point scenario. ................................................................. 48
3.2 RLNC encoded packets ....................................................................... 49
3.3 Decoding error probability of RLNC in $\mathbb{F}_4$ and $\mathbb{F}_{256}$ ................. 50
3.4 Structure of hybrid serial concatenated network code encoders ............ 51
3.5 Structure of the input data matrix for LT encoder in a hybrid serial concatenated network code ............... 54
3.6 Decoding error probability gain of hybrid serial concatenated network codes with LT outer codes in respect of RLNC .................. 55
3.7 Robust Soliton distribution with $\delta = 0.001$ and $c = 0.1$. .......... 57
3.8 Robust Soliton distribution with $\delta = 0.1$ and $c = 0.1$. ............ 58
3.9 Robust Soliton distribution with $\delta = 4$ and $c = 0.1$. ............ 59
3.10 Decoding error probability of HSCNC with LT code with $\delta = 0.001$ and $c = 0.1$, and with RLNC with $\mathbb{F}_4$ .......... 61
3.11 Decoding error probability of HSCNC with LT code with $\delta = 0.1$ and $c = 0.1$, and with RLNC with $\mathbb{F}_4$ .................. 61
3.12 Decoding error probability of HSCNC with LT code with $\delta = 4$ and $c = 0.1$, and with RLNC with $\mathbb{F}_4$ .................. 62
3.13 Decoding error probability of HSCNC with LT code with $\delta = 0.001$ and $c = 0.1$, and with RLNC with $\mathbb{F}_{256}$ .......... 63
3.14 Decoding error probability of HSCNC with LT code with $\delta = 0.1$ and $c = 0.1$, and with RLNC with $\mathbb{F}_{256}$ .................. 64
3.15 Decoding error probability of HSCNC with LT code with $\delta = 4$ and $c = 0.1$, and with RLNC with $\mathbb{F}_{256}$ .................. 65
3.16 Encoding and decoding time of HSCNCs with LT outer codes. .... 66
3.17 Decoding error probability gain of hybrid serial concatenated network codes with RS outer codes in respect of RLNC .................. 68
3.18 Decoding error probability of HSCNC with RS code and with RLNC with $R = 0.94$ and $\mathbb{F}_4$ .......... 70
3.19 Decoding error probability of HSCNC with RS code and with RLNC with $R = 0.91$ and $\mathbb{F}_4$ .......... 70
3.20 Decoding error probability of HSCNC with RS code and with RLNC with $R = 0.94$ and $\mathbb{F}_{256}$ .......... 71
3.21 Decoding error probability of HSCNC with RS code and with RLNC with $R = 0.91$ and $\mathbb{F}_{256}$ .......... 72
3.22 Encoding and decoding time of HSCNCs with FRS outer codes. .... 73
3.23 Decoding error probability of HSCNC with LDPC code and with RLNC over $\mathbb{F}_4$ .......... 74
3.24 Decoding error probability of HSCNC with LDPC code and with RLNC over $\mathbb{F}_{256}$ .......... 75
3.25 Encoding and decoding time of HSCNCs with LDPC outer codes. .... 76
3.26 Encoding and decoding time of HSCNCs with Raptor (LDPC pre-coding) outer codes. .......................................................... 78
3.27 Decoding error probability gain of hybrid serial concatenated network codes with Raptor outer codes (RS pre-coding) in respect of RLNC . . . 79
3.28 Decoding error probability of HSCNC with Raptor codes ($\delta = 0.001$) and with RLNC over $\mathbb{F}_4$ ................................. 80
3.29 Decoding error probability of HSCNC with Raptor codes ($\delta = 0.1$) and with RLNC over $\mathbb{F}_4$ ........................................ 81
3.30 Decoding error probability of HSCNC with Raptor codes ($\delta = 4$) and with RLNC over $\mathbb{F}_4$ ........................................ 82
3.31 Decoding error probability of HSCNC with Raptor codes ($\delta = 0.001$) and with RLNC over $\mathbb{F}_{256}$ ................................. 83
3.32 Decoding error probability of HSCNC with Raptor codes ($\delta = 0.1$) and with RLNC over $\mathbb{F}_{256}$ ........................................ 83
3.33 Decoding error probability of HSCNC with Raptor codes ($\delta = 4$) and with RLNC over $\mathbb{F}_{256}$ ........................................ 84
3.34 Encoding and decoding time of HSCNCs with Raptor outer codes (RS pre-coding). .......................................................... 85

4.1 Example of MIHF communication model. ................................. 90
4.2 Example of MIHF communication model for hard VHOs. ............... 91
4.3 Example of MIHF communication model for soft VHOs. ............... 91
4.4 Scenario of simulation of vertical handovers between WiMAX and WLAN with the MIH protocol. ........................................ 93
4.5 Burst erasure probability versus number of mobile nodes. ............... 94
4.6 Burst erasure probability versus transmission rate. ...................... 95
4.7 EC-MIH reference model ..................................................... 96
4.8 Logic scheme of EC-MIH protocol and EC sub-layer. .................... 100
4.9 EC-MIH protocol frame format. ........................................... 101
4.10 EC-MIH header format. ..................................................... 103
4.11 Topology of simulations of performance of MIH and EC-MIH protocol. 104
4.12 Burst erasure probability versus number of mobile nodes. ............. 106
4.13 Burst erasure probability versus transmission rate. ..................... 106
4.14 Throughput versus time of MIH protocol during vertical handovers given the burst erasure probability. ................................ 110
4.15 Throughput versus time of MIH protocol during hard vertical handovers for different handover time. ............................................. 111
4.16 Average throughput versus coding rate of EC-MIH protocol with HSCNCs. 112
4.17 Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with LT outer codes. ................................. 113
4.18 Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with LT outer codes. ................................. 115
4.19 Comparison of vertical handover time of MIH and EC-MIH with HSCNCs (LT outer codes). ................................................. 116
4.20 Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with RS outer codes. ................................. 117
4.21 Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with RS outer codes. ................................. 118
4.22 Comparison of vertical handover time of MIH and EC-MIH with HSCNCs (RS outer codes). ................................................. 119
4.23 Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with Raptor outer codes (RS pre-coding stage). ..... 120
4.24 Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with Raptor outer codes (RS pre-coding stage). ..... 121
4.25 Comparison of vertical handover time of MIH and EC-MIH with HSCNCs (Raptor outer codes with RS pre-coding). ........................ 122
4.26 Average throughput gain versus burst erasure probability of EC-MIH protocol with HSCNCs. ............................................... 123
4.27 Average throughput gain versus burst erasure probability of EC-MIH protocol with HSCNCs. ............................................... 124
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>Third Generation Partnership Project</td>
</tr>
<tr>
<td>ARQ</td>
<td>Automatic Repeat Request</td>
</tr>
<tr>
<td>BATS</td>
<td>Batched Sparse</td>
</tr>
<tr>
<td>BCH</td>
<td>Bose-Chaudhuri-Hocquenghem</td>
</tr>
<tr>
<td>BEC</td>
<td>Burst Erasure Channel</td>
</tr>
<tr>
<td>BP</td>
<td>Belief-Propagation</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>EC-MIH</td>
<td>Enhanced-Coded MIH</td>
</tr>
<tr>
<td>EC-MIHF</td>
<td>Enhanced-Coded MIHF</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FRS</td>
<td>Folded Reed-Solomon</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>HARQ</td>
<td>Hybrid Automatic Repeat Request</td>
</tr>
<tr>
<td>HSCNC</td>
<td>Hybrid Serial Concatenated Network Code</td>
</tr>
<tr>
<td>IE</td>
<td>Information Element</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LDPC</td>
<td>Low-Density Parity-Check</td>
</tr>
<tr>
<td>LNC</td>
<td>Linear Network Coding</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LT</td>
<td>Luby Transform</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MDS</td>
<td>Maximum Distance Separable</td>
</tr>
<tr>
<td>MICS</td>
<td>Media-Independent Command Service</td>
</tr>
<tr>
<td>MIES</td>
<td>Media-Independent Event Service</td>
</tr>
<tr>
<td>MIIS</td>
<td>Media-Independent Information Service</td>
</tr>
<tr>
<td>MIH</td>
<td>Media-Independent Handover</td>
</tr>
<tr>
<td>MIHF</td>
<td>Media-Independent Handover Function</td>
</tr>
<tr>
<td>MN</td>
<td>Mobile Node</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmission Unit</td>
</tr>
<tr>
<td>NEC</td>
<td>Network Error Correction Coding</td>
</tr>
<tr>
<td>PoA</td>
<td>Point of Attachment</td>
</tr>
<tr>
<td>PoS</td>
<td>Point of Service</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of service</td>
</tr>
<tr>
<td>RLNC</td>
<td>Random Linear Network Coding</td>
</tr>
<tr>
<td>RNC</td>
<td>Radio Network Controller</td>
</tr>
<tr>
<td>RS</td>
<td>Reed-Solomon</td>
</tr>
<tr>
<td>SAP</td>
<td>Service Access Point</td>
</tr>
<tr>
<td>SCTP</td>
<td>Stream Control Transmission Protocol</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TLV</td>
<td>Type-Length-Value</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>UTRAN</td>
<td>Universal Terrestrial Radio Access Network</td>
</tr>
<tr>
<td>VHO</td>
<td>Vertical Handover</td>
</tr>
<tr>
<td>WiMAX</td>
<td>Wireless Interoperability for Microwave Access</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Chapter Outline: This chapter introduces the key contribution considered in this work, addressing network coding for efficient vertical handovers. In this context, we first provide an overview on vertical handovers and highlight existing issues and challenges. A pivotal feature is the IEEE 802.21 standard [2] to perform media-independent vertical handovers for technology agnostic systems. Moreover, we discuss burst erasure codes and network coding with focus on their benefits in vertical handover scenarios. Thereafter we present the current research challenges in the design of efficient vertical handover schemes, and describe how burst erasure-correcting codes and network coding schemes for burst erasure recovery can play an important role towards providing effective vertical handover solutions for next generation networks. Finally, the thesis contribution to state-of-the-art and the organisation are outlined.

1.1 Overview on Vertical Handovers

Currently the number of mobile handsets equipped with multiple interfaces is continuously increasing. The possibility of performing Vertical Handover (VHO) among different access technologies led to new design challenges in heterogeneous networks and mobility management. In order to address these challenges, many different approaches were proposed in the last decade, and now vertical handover still continues to play a pivotal role and is considered an essential ingredient of 5G mobile networks.
1.1. Overview on Vertical Handovers

Figure 1.1: General taxonomy of proposed solutions to vertical handovers. The initial technology-dependent solutions are in the lighter shapes while latest technology-independent solution (IEEE 802.21) is in the darker shape.

There are two ways of performing a VHO: the first is called *break-before-make* (hard VHO) since only one link at a time is active; this mechanism is particularly suitable for delay-tolerant communication traffic. The second is called *make-before-break* (soft VHO) because more than one link is active at a time during handover operations. For example, soft VHO mechanism is suitable for Code Division Multiple Access (CDMA) applications: in fact, due to the properties of the CDMA signalling scheme, it is possible for a CDMA mobile phone to simultaneously receive signals from two or more radio base stations that are transmitting the same bit stream (using different transmission codes) on the different physical channels in the same frequency bandwidth.

Figure 1.1 shows a simple taxonomy of the proposed solutions to vertical handovers. The solutions are mainly grouped into technology-dependent and technology-independent.

The research community initially focused on the design of technology-dependent VHO mechanisms. Many articles studied specifically how to efficiently perform VHOs among Wireless Local Area Network (WLAN) (IEEE 802.11), Wireless Interoperability for Microwave Access (WiMAX) (IEEE 802.16) and Third Generation Partnership Pro-
1.1. Overview on Vertical Handovers

ject (3GPP) cellular networks. The European Telecommunications Standards Institute (ETSI) made the first standardisation effort toward the categorisation of all interworking architectures between WLAN and 3GPP networks in 2001. Next, more recent standardisation efforts by 3GPP have tried to propose an interworking categorisation based on the level of interworking between the two networks. So, all the solutions to manage VHO procedures between WLAN and 3GPP networks can be correctly grouped into three categories [1]: loose coupling, tight coupling and very tight coupling. The first considers no integration between the networks, which only shares the subscription. This is not adequate for tight performance guarantee. The second provides some level of integration between the networks, and it is mainly applicable when WLAN is part of operator’s network. The third sees WLAN as another access technology for 3GPP network. This is quite complicated and only few solutions have been proposed by the research.

The proposed approaches to perform vertical handovers between WLAN and WiMAX mainly focused on providing common Quality of service (QoS) mapping in the integrated network. However, this is still challenging issue especially because the two technologies use different Media Access Control (MAC) metrics.

Finally, the technology-dependent solutions for VHOs between 3GPP and WiMAX resulted to have limitations due to abrupt disconnections on the source network side. That has become a cause of higher packet loss. A deeper review of these aspects is provided in section 2.1.

The previous discussion highlighted the fact that the interworking systems for VHOs present high complexity of design and implementation, while they leave some problems open. In order to avoid this and address the lack of scalability due to the presence of many different VHO frameworks for each couple of technologies, the IEEE released its standard IEEE 802.21 MIH Services [2, 3] for technology-independent vertical handovers in 2009. In fact, the main purpose of the standard is to provide a general framework to enable handovers between heterogeneous technologies (IEEE and 3GPP) without service interruption, hence improving the mobile user experience. This framework allows higher layers to interact with lower layers to provide session continuity.
1.1. Overview on Vertical Handovers

without dealing with the specifics of each technology. Moreover, the standard defines a protocol, which acts as a glue between the Internet Protocol (IP)-centric world developed in Internet Engineering Task Force (IETF), and the reference scenarios for future mobile networks currently being designed in 3GPP. While the IETF does not address specific layer 2 technologies, the interest of 3GPP in noncellular layer 2 technologies, such as WLAN, is restricted to its integration into cellular environments. IEEE 802.21 provides the missing, technology-independent, abstraction layer able to offer a common interface to upper layers, thus hiding technology-specific primitives. This abstraction can be exploited by the IP stack (or any other upper layer) to better interact with the underlying technologies, ultimately leading to improved handover performance.

IEEE 802.21 standard guarantees reliability of its MIH protocol through ARQ. Nevertheless, when losses increase, this technique can maintain reliability while becoming less efficient, and acknowledgements and retransmission timers contribute to degrade performances. The way to try to mitigate these drawbacks comes from the deployment of Hybrid Automatic Repeat Request (HARQ): this is a variation of ARQ, which combines ARQ and FEC. However, the use of pure FEC can outperform ARQ by guaranteeing reliability while reducing latency. Moreover, it has been demonstrated [4, 5, 6] that pure FEC based on RLNC outperforms HARQ and subsequently ARQ. Hence, let’s briefly introduce error-correcting codes and network coding.

1.1.1 Error-Correcting Codes for Burst Erasure Channels

When a source communicates with a destination, the information is sent via a communication channel. However, real channels always introduce some noise and the sink does not receive what was sent by the source. So, the main aim is to manipulate the source message in some ways in order to enable the recovery of the original information at the receiver.

Error-correcting codes help reduce the impact of channel errors on the communications. The basic idea is to add some redundancy to the original information in the time domain to mitigate the effect of the error probability of the channel on the correct reception
of the message. However, this comes at the price of reduction in data throughput. As an example, let us imagine that four redundant symbols are sent for each source symbol: that reduces the transmission rate of \( \frac{1}{4} \). The greater the redundancy, the higher is the confidence level and the lower is the communication rate. So, according to Shannon’s theorem, a source must increase the complexity of the communication scheme to reliably send its message at a rate close to channel capacity.

A channel may introduce erasures as well as errors. An erasure is a message component that is marked as unknown. It differs from an error, which is a message component that is received but is incorrect. There are several conditions that can cause erasures such as buffer overflows, congestions, excessive delays, disconnections and received symbols too weak to recognise. Furthermore, the errors and erasures can occur in a burst under some conditions. If this happens, the channel is called with memory since the erasures (errors) are not independent.

Currently there are some particular coding schemes that are suitable for erasure (and burst erasure) correction. In the family of block cyclic codes, Reed-Solomon (RS) code [7] represents a very powerful coding scheme to help recover random and burst erasures. Low-Density Parity-Check (LDPC) [8] codes are linear block codes whose parity-check matrix can be optimised for burst erasure recovery. In order to correct erasures, novel rateless coding schemes were proposed called fountain codes. The real implementation of these codes are Luby Transform (LT) codes [9, 10] and Raptor codes [11, 12]. The detailed characteristics of these codes are described in section 2.3.

1.1.2 Network Coding

Network coding has been demonstrated to assist with reduction in acknowledgement overhead [13], and to improve protocol (i.e. TCP) [14, 15] and communication performance in heterogeneous networks [15]. In order to visualise the evolution of network coding from its inception, figure 1.2 depicts a visual technology roadmap till the present day, in which labels are used to identify the most significant articles in the major areas of this field.

The publication of [16] is the commencement of network coding theory: in fact, this
article is the first to refer to this novel research field as network coding. Ahlswede et al. revealed a new research topic by studying the problem of characterising the coding rate region of a multicast scenario. The main result of the authors consisted in a max-flow min-cut theorem, which interpreted the flow of information as something not to be merely routed and replicated anymore. However, that was not only a starting point but also a point of arrival: the work in [16] took advantage of many concepts in information theory, distributed coding and distributed data storage systems, developed during the previous years. First, the geometric framework and the set theory, which were developed to simplify the solution of information theoretic problems, improved the methods to prove converse coding theorems and to calculate coding rate regions. Next, the models used for distributed coding and distributed data storage systems provided special instances to develop the general one deployed by [16].

Subsequently, Li et al. [17] described an optimal solution achieving the max-flow min-cut bound for directed acyclic graphs: the optimum was obtained by applying a linear code multicast, a network code obtained by linearly combine information by using coefficients chosen from a finite field. That definition of linear network code mainly employed tools from graph theory and algebra of vectors. This article [17] started the formulation of the concepts of the theoretic framework used by deterministic network coding. Among the works contributing to the general theory of deterministic Linear Network Coding (LNC), the new approach of [18] opened another way for network coding theory: instead of using some elements from algebra as in [17], Koetter et al. [18] developed a completely algebraic framework by making connections with algebraic geometry and matrix theory. The fundamental results in [18] prepared the fertile ground for the formulation of RLNC. This family of network codes has the main characteristic of randomly choosing the coefficients of linear combinations making them suitable for dynamic environments. However, such codes are subject to one key drawback: an error decoding probability, which depends on the size of the finite field chosen for the code.

The investigation into network flows and LNC provided the general theoretic background to apply network coding on directed acyclic scenarios: combination networks raised particular interest because of the various areas of application. Nevertheless, Z. Li and B. Li [19] started the analysis of behaviours and benefits of network coding in
Figure 1.2: The graph depicts the evolution of network coding theory and its main areas of research. The labels identify the most significant articles, and the blue circles underline the areas, which have attracted most of the research interest.
undirected networks. Side by side, convolutional network codes were proposed as a better solution than classical block network codes, in directed cyclic graphs: by taking into account cycles, the information in the network experiences delays, that are different from zero. So, the time dependence of the model introduces a trellis structure, which becomes similar to a trellis diagram of a convolutional code. Lately, S.-Y. R. Li and Q. T. Sun [20] demonstrated that acyclic LNC theory can be extended to cyclic scenarios: precisely, convolutional network codes come to be an instance of network codes defined via commutative algebra.

Network coding theory enlarged its area of application when R. Yeung and N. Cai [21, 22] showed that network codes represent a generalisation of classical error-correcting codes. The attention in Network Error Correction Coding (NEC) was due to its inherent strengths in correcting random errors, erasures and errors introduced by malicious nodes. The main characteristic of the network extension of error correction codes is that redundancy is in space domain instead of in time domain.

After its foundation, the study of network coding focused on other fundamental theoretic topics to better understand its performance such as the coding capacity achievable in different scenarios compared with the one achievable with classical store-and-forward routing, and the conditions under which a network is solvable\(^1\). The link with matroid theory and the results obtained in information theory about information inequalities provided researchers a strong background to face the capacity and solvability issues of network coding.

Complementing the pure theory of network coding, a branch of investigation was focusing on the algorithmic implementation of network codes [24, 25]. The complexity of the algorithms is influenced by the size of the finite field required by the network code and depends on the number of sinks in the communication. Other elements affecting the complexity are also the number of edges and the transmission rate of the source. The deployment of many tools from combinatorics and graph theory helped the design

\(^1\text{A multicast network is solvable [23] with respect to the alphabet if the encoding operations can be assigned in such a way that the source messages can always be recovered at each destination node, using decoding operations. The encoding and decoding operations in a network collectively constitute a 'code'.}\)
of algorithms with reduced complexity.

Recently, variable-rate network coding [26] appeared on the scene. This paradigm consists in using different rates for different sessions in order to enhance fixed-rate linear network codes.

1.2 Research Challenges

The aim of this thesis is to achieve efficient VHOs in terms of providing greater resiliency and reduced latency, both pivotal requirements for next generation (5G) broadband services. An efficient and reliable protocol has been proposed to accomplish this goal by integrating classical erasure-correcting codes and RLNC as part of the MIH protocol. Hence, the work of this thesis connects different fields of telecommunications in order to harness the benefits of each technology to address the challenges in VHO. In the following sections the challenges associated with vertical handovers, network error correction codes and burst-erasure codes are discussed.

1.2.1 Challenges in Vertical Handovers

The capability to simultaneously maintain connections with several wireless technologies and seamlessly move between them, has been mentioned among the key aspects of future 5G mobile networks. In this context, IEEE 802.21 standard provides a media-independent framework to efficiently allow VHOs among different technologies. However, MIH protocol has weaknesses that can negatively affect handover procedures. Latency and losses represent the main challenges during handover operations. IEEE 802.21 standard is designed to minimise handover delay and losses. However, the adverse surrounding conditions during a handover can increase the loss probability. Reference [27] showed that MIH packet loss increases when the transmission rate and the number of Mobile Node (MN) in the network increase. Hence, if a candidate network provides a transmission rate three/four times greater than the one of the current network, the mobile user experiences an increased loss. Side by side, MIH protocol uses acknowledgement service and retransmissions of lost frames to provide reliability. This
makes conditions even more complicated for the protocol. The acknowledgement service and retransmissions require time, which increases the handover delay. Nevertheless, Marques et al. [27] also represented how VHO delay affects losses. So, it appears that several MNs increase losses that increase delay that further increases the loss probability. Moreover, other adverse operational conditions can affect VHO procedures such as periods of low signal, fast mobility, congestions, and time to switch interfaces and networks.

MIH protocol messages are sent over the data plane by using a suitable transport mechanism at both layer 2 and layer 3. The main protocols mentioned by the standard are Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Stream Control Transmission Protocol (SCTP) over IP. TCP and SCTP implement acknowledgements while UDP has no acknowledgement service. This means that the formers increase the communication delay in case of erasures. Especially for TCP, the size of the congestion window can be limited both by the acknowledgement service of MIH protocol and by losses of TCP packets. On the other side, the latter cannot improve MIH reliability since it does not provide any acknowledgement policy.

1.2.2 Challenges in Network Error Correction Codes

Network coding can be used to mitigate packet loss on a lossy link, just like any other erasure-correcting code. If a good estimate is available surrounding the error probability, then a certain amount of redundancy packets can be generated pro-actively to combat packet losses. The redundancy has to be carefully chosen not to significantly reduce the information rate of the communication. The disadvantage of RLNC in comparison with ‘classical’ erasure codes is that the minimum distance of RLNC is a random variable and not a deterministic value. Hence, it is important to monitor this parameter to guarantee the error-correcting capability required by the application.

Random linear network coding gives the advantage of a noncoherent implementation of network coding. On the other hand, the price to pay is a reduction in the successful decoding probability that mainly depends on the size of the network and on the size of the finite field of the code. In particular, the smaller is the finite field, the greater
1.2. Research Challenges

is the failure probability. Side by side, the increase of the finite field augments the complexity of the code. Hence, there is a trade-off between the failure probability and the complexity of the encoding/decoding operations.

As just mentioned, the size of the finite field affects the complexity of the code. This also becomes an important parameter that influences the communication delay. In fact, the operations in greater Galois fields require more time and the encoding/decoding throughput can decrease.

A generation is a set of packets sent by the same source, accomplished by labelling packets’ headers with a generation number. The operation of packet tagging (appending random coefficients of the linear combinations) has a cost in terms of overhead. Especially, the increase in both the generation size and the field size augments the overhead of the communication.

When random linear combinations of packets are transmitted, the issue of how to acknowledge data arises. That has been solved [14] by acknowledging not the reception of all the packets, but only the one of linear independent packets. Anyway, this effective solution can give some drawbacks if also the acknowledgements are subject to losses: in fact, it can raise the decoding delay.

The problems related to the design of erasure correction RLNC which have been highlighted in this subsection, are fully explained and analysed in section 2.4.

1.2.3 Challenges in Burst Erasure Codes

This thesis deploys the most common erasure coding schemes such as RS codes, LDPC codes, LT codes and Raptor codes. All of them raise design issues and challenges.

RS codes are very well known and investigated codes. However, they tend to be relatively complex, especially if concatenated with other coding schemes. At a fixed size of the finite field, the complexity of RS codes mainly depends on the size of the redundancy. Hence, the aim is to optimise the error-correcting capability at the lowest possible redundancy.
LDPC codes are efficient codes but require particular attention to the design of their parity-check matrix. So, it is fundamental to find an optimal structure of the parity-check matrix to obtain the highest erasure-correcting capability while reducing the complexity and the redundancy.

Fountain codes are very efficient and effective codes for erasure correction. In case of LT codes, their design depends on the set of the parameters of the Robust Soliton distribution [12]. So, it is important to draw a distribution that optimises the error-correcting capabilities of the code for an assigned erasure channel. Moreover, LT codes are demonstrated to be very effective for input data stream of size greater than $10^6$. That enforces the importance of the design of the Robust Soliton distribution. Regarding Raptor codes, the issues related to the distribution of LT codes come together with the ones related to the presence of a pre-coding stage. In fact, the error-correcting capability and the efficiency of Raptor codes are fully determined by an efficient combination of the pre-coding stage with LT codes.

1.3 Thesis Contribution

This thesis provides the following contributions to solve the aforementioned challenges.

1.3.1 Forward Error Correction for Efficient Vertical Handovers

The first aim of this thesis is to go beyond the current implementation of MIH by proposing EC-MIH to provide more effective VHOs. The main motivation towards this new incarnation is to provide reliable communications during VHOs by deploying FEC to overcome the need for any acknowledgement service: this can significantly reduce handover time. In fact, the reliability is provided by built-in coding operations that can mask the losses due to VHO operations: in fact EC-MIH frames are coded frames. The protocol provides a new architecture and logical structure that efficiently integrate the coding operations in the architecture of IEEE 802.21 standard. These coding operations consider that the encoder periodically obtains an estimation of packet loss caused by the current handover procedure. This estimation is provided by a logical block, which
gets and processes information about characteristics and status of the vertical handover. However, as explained in detail in chapter 4, the design of the estimation algorithm is out of the scope of this thesis. Moreover, EC-MIH protocol modifies the MIH frame structure to integrate these new functionalities.

1.3.2 Novel Hybrid Coding Scheme for Forward Error Correction

We propose new optimal coding schemes based on network coding to optimise EC-MIH protocol performance. These kind of schemes are classified as HSCNC. The term *hybrid* means that the serial concatenation involves a network error-correcting code and an erasure code. The NEC scheme used in this work is RLNC which is noncoherent, so it can keep the implementation topology-independent. In particular, HSCNCs exploit systematic RLNC as inner code. The use of systematic version allows the outer erasure code to effectively recover lost frames if RLNC fails to decode. Furthermore, the complexity of systematic RLNC is significantly less than the one of non-systematic RLNC.

The HSCNCs are applied on an end-to-end basis and not applied at each link as 'classical' error-correcting codes. This characteristic decreases the complexity of HSCNC compared to the one of product network codes, while guaranteeing equal or greater erasure protection.

This work studies and analyses the error-correcting capability and the encoding/decoding time of HSCNCs. In particular, the outer codes under consideration are: LT codes, RS codes (especially Folded Reed-Solomon (FRS)), LDPC codes and Raptor codes. Hence, the design of effective HSCNCs requires the careful selection of several coding parameters to optimise the concatenation with systematic RLNC.

First, we design the optimal robust Soliton distribution for LT outer codes. In fact, the thesis proposes particular shapes for the robust Soliton distribution to maximise the performance for small size data streams. Second, we show how to set the parameters of FRS codes. Next, we explain why LDPC codes are not optimal as outer codes in our context. Finally, we study the set of parameters, that optimise the performance of
1.3. Thesis Contribution

Raptor outer codes. Beyond the scheme proposed by the literature with LDPC pre-coding stage, we suggest the exploitation of FRS codes for pre-coding. Finally, we show which concatenated coding schemes are effective and efficient for EC-MIH protocol.

1.3.3 Hybrid Coding for EC-MIH Protocol

The hybrid coding schemes, that are integrated into EC-MIH protocol, are the ones which can provide reliability for coding rate greater than 0.5 with reasonable complexity. In particular, the simulation results in this thesis show that EC-MIH protocol provides reliable communications more efficiently than legacy MIH. In order to achieve those results, a Gilbert model of VHOs with MIH is developed to allow a meaningful evaluation of EC-MIH protocol. In addition, functions to model the packet loss during VHOs with MIH protocol are provided: this new functions are obtained by fitting simulated data of a proprietary simulator [28], and are used in order to model MIH frame loss as a burst erasure channel.

The study considers the use of TCP as a transport protocol. The efficiency of EC-MIH protocol is principally measured in terms of throughput achieved by TCP when MIH and EC-MIH protocol experience losses due to VHO operations. The overall results show that EC-MIH is in general more efficient and effective than the original MIH.
1.3. Thesis Contribution

List of publications

The following papers are the outcome of my work during the Ph.D.:

Journal papers:


Book chapters:


Conference papers:


1.4 Organisation of the Thesis

The rest of the thesis is organised as follows:

Chapter 2 presents the state-of-the-art on existing approaches to design technology-dependent vertical handovers. Moreover, it provides the theoretical background on IEEE 802.21 standard, MIH protocol, burst erasure channels and error-correcting codes for burst erasure channels. This background is important to understand the achievements of this work. Furthermore, the description of this chapter is useful to introduce the mathematical notation that will be used throughout this thesis.

Chapter 3 describes the theory and the structure of HSCNCs. It provides analysis and simulations of HSCNC performance by considering different combinations of coding schemes. Moreover, it shows how to set parameters to optimise their performances. Furthermore, it investigates the encoding/decoding throughput and the complexity of different HSCNCs.

Chapter 4 presents the characteristics of the newly proposed EC-MIH protocol such as the system requirements and functions, architecture, logical functionalities and its packet structure. In parallel, polynomial functions are calculated in order to model VHOs as burst erasure channels. Next, this chapter provides the simulation results of the EC-MIH protocol for VHOs by integrating the HSCNCs from chapter 3. The holistic performance of EC-MIH is calculated in terms of throughput experienced by TCP and handover time.

Chapter 5 concludes the thesis by highlighting the outcomes of this work and suggests some directions for further work.
Chapter 2

IEEE 802.21 Standard and Error-Correcting Codes for Burst Erasure Channels

Chapter Outline: This chapter begins with a description of the existing approaches to VHOs, according to the technologies involved, in order to show the path that led to IEEE 802.21 standard. So, it provides detailed background related to IEEE 802.21 standard and MIH protocol. Next, the discussion moves to burst erasure channels and erasure-correcting codes. First, it describes the model that is of particular interest for analysing losses of MIH during VHOs. Next, it explains the main characteristics of RS and FRS codes, the general theory of fountain codes (LT and Raptor codes) and their relevance. In particular, we highlight the importance of the distribution design in the codes to influence performance. The chapter describes an efficient LDPC coding scheme for burst erasure correction. Finally, we introduce the theory of NEC codes with their advantages and drawbacks. Furthermore, the chapter also highlights the open challenges associated with the different coding schemes, that this work has addressed. We also introduce the mathematical notations used in the thesis for analysis and simulations in the remaining chapters.
2.1 Existing approaches to Vertical Handovers

As briefly introduced in the previous chapter, the research on VHOs initially focused on the design of mechanisms based on the particular technologies involved. As we said, we can identify three main categories [1] to classify all the solutions to perform VHOs between WLAN and 3GPP networks: loose coupling, tight coupling and very tight coupling.

Loose coupling solution is the dominant and most widely used among the proposed architectures. It indicates a means of interconnecting independently the two networks by only utilising a common subscription. The common factor for all the proposed architectures in this category is the use of Mobile IP as the basic instrument for inter-system mobility and for integration. This approach is adequate for service continuity, but not for tight performance guarantees. In this solution, interconnected networks are considered as independent networks concerning the handling of data traffic.

Tight coupling solution shows WLAN as another access network to the cellular core network; thus, both data and signalling traffic are transferred through the cellular network. This solution is considered to apply in cases where a WLAN is directly attached to a 3GPP core network component and affecting the functionality of this component.

Very tight coupling schemes focus on interworking at the Universal Terrestrial Radio Access Network (UTRAN) level and, more precisely, on incorporating Radio Network Controller (RNC) or lower cellular network entities functionality into WLAN components. This approach improves handover performance but the implementation complexity is considerably increased. The re-use of part of the cellular functionality makes the application of tight and very tight coupling solutions more appropriate for cellular operators that deploy their own WLANs.

On the other side, many works have tried to design efficient architectures to perform VHOs between WLAN and WiMAX networks [29, 30]. Providing QoS in the integrated IEEE 802.16/WiMAX and IEEE 802.11/WLAN network is a challenging issue. This difficulty especially arises from the fact that the two technologies integrate differ-
2.1. Existing approaches to Vertical Handovers

Existing approaches to Vertical Handovers typically consider different MAC metrics, provide different transmission rates and different support for QoS. So, the integration of WLAN and WiMAX has to take into account a QoS mapping procedure. Hence, Gakhar et al. [31] proposed an architecture to provide end-to-end QoS in an integrated WiMAX/WLAN network. This proposal strives to map the QoS requirements of an application originated in an IEEE 802.11e network to those of a serving WiMAX network.

The research on QoS mapping between WLAN and WiMAX classes focuses, in general, on a static translation scheme. But this approach has some drawbacks such as it does not take into account current network load and therefore does not benefit from adaptation mechanisms, and the mapping is network-specific thus neither flexible nor scalable. An efficient solution came from the IETF in the form of Mobile IP. For the reader interested in upper layers protocols for mobility management in all-IP networks, it is useful to see [32].

Finally, part of the research community also worked on the integration of 3GPP and WiMAX technologies. For the integration of mobile WiMAX and 3GPP networks, the 3GPP Evolved Packet Core (EPC) provides internetworking functionalities according to Release 8 of the 3GPP specifications [33]. The internetworking functionalities include authentication of the User Equipment (UE), access network discovery, QoS consistency, and seamless handover. The research on the design of interworking WiMAX-3GPP systems for seamless VHOs, has followed three main guidelines. First, the VHO solution should not require significant change to networking systems that are already deployed. Second, the VHO process needs to be optimised to minimise packet loss and execution time. Third, an IP-based solution is preferred for internetworking of heterogeneous access networks and potential extensions. Formerly proposed VHO solutions introduced several logical interfaces and entities: however, these solutions still have limitations due to abrupt disconnection on the source network side that results in packet loss. Hence, Song et al. [34] proposed an additional logical entity in order to overcome these problems.
2.1. Existing approaches to Vertical Handovers

2.1.1 IEEE 802.21: Media-Independent Handover Services

In 2009, IEEE provided a technology-independent solution to VHOs. This latest solution released a general framework to guarantee efficient and reliable VHOs between any kind of technology, either wireless or wired.

Figure 2.1 shows IEEE 802.21 reference model. There are three kinds of network entities:

- MIH Point of Attachment (PoA): This is the endpoint of a layer 2 link that includes the mobile node as the other endpoint.

- MIH Point of Service (PoS): This is a network entity that exchanges MIH messages with the mobile node. Note that a mobile node may have different PoSs as it may exchange messages with more than one network entity.

- MIH non-PoS: This is a network entity that does not exchange MIH messages with the mobile node. Note that a given network node may be a PoS for a mobile node with which it exchanges MIH messages and a non-PoS for a network node for which it does not.

The communication between these network entities is possible via five reference points:
2.1. Existing approaches to Vertical Handovers

- Reference point R1 is used by the mobile node to communicate with its PoA. Among other purposes, it may be used by the mobile node to gather information about the current status of its connection.

- Reference point R2 is used by the mobile node to communicate with a candidate PoA. It may be used to gather information about candidate PoAs before making a handover decision.

- Reference point R3 is used by the mobile node to communicate with an MIH PoS located on a non-PoA network entity. It may be used by a network node to inform the mobile node about the different IP configuration methods in the network.

- Reference point R4 is used for communications between an MIH PoS and an MIH non-PoS. This reference point is typically used when an MIH server that is serving a mobile node (the PoS) needs to ask for information from another MIH server (the non-PoS).

- Reference point R5 is used between two different MIH PoSs located at different network entities.

Figure 2.2 represents MIH architecture and the types of Media-Independent Handover Function (MIHF) relationships. The communication between the MIHF and MIHF users, lower layers or remote entities, is based on a number of service primitives that are called Service Access Point (SAP). There are three kind of SAPs: MIH_SAP, MIH_LINK_SAP and MIH_NET_SAP.

The MIH_SAP is a media-independent interface between the MIHF and upper layers. The upper layers need to subscribe with the MIHF as users to receive MIHF-generated events. The same also happens for link-layer events that originate at lower layers, but that are passed on to MIHF users via the MIHF. MIHF users directly send commands to the local MIHF using the service primitives of the MIH_SAP. Next, MIH_LINK_SAP is a media-dependent interface between the MIHF and lower layers media-specific protocol stacks of technologies. For different link-layer technologies, media-specific SAPs provide the functionality of MIH_LINK_SAP. Finally, MIH_NET_SAP is a media-dependent
2.1. Existing approaches to Vertical Handovers

interface of the MIHF that provides transport services over the data plane on the local node, supporting the exchange of MIH information and messages with remote MIHFs. The full list of the primitives that each SAP support, can be found in [2, Subsec. 7.2].

The standard defines three services that comprise the MIHF service: Media-Independent Event Service (MIES), Media-Independent Command Service (MICS) and Media-Independent Information Service (MIIS). These services facilitate handovers between heterogeneous access links. IEEE 802.21 supports handover initiated by either the network or mobile terminals; hence, events related to handovers can be originated at MAC or MIHF layer located in the node or at the PoA.

The MIES can be divided in two categories, link and MIH. Link events are generated within the link layer and received by the MIHF. Events that are propagated by the MIHF to MIHF users are called MIH events. Note that link events propagated to upper layers become MIH events. Entities able to generate and propagate link events are the defined IEEE 802.x, 3GPP and 3GPP2 MIH_LINK_SAP interfaces.

The MICS is the group of commands sent from higher layers to the lower layers in order to check the status of links and configure the devices to optimise handover policies. The reception of a command may cause an event so that the consequences of a command can be tracked by network entities. There are remote and local commands. The formers can be used to force a terminal to perform handover during a network-assisted VHO. The others can allow the user to configure the lower layers. The MICS consists in MIH
2.1. Existing approaches to Vertical Handovers

Media-Independent Information Service (MIIS) is a tool used by MIHF to get network information within a geographical area to help VHOs. This service is based on Information Element (IE). The IEs give relevant information for network selection and for VHO completion.

Reference [2] fully lists and describes the event, command and information services that IEEE 802.21 makes available for VHO procedures.

Media-Independent Handover Protocol

According to IEEE 802.21 [2], MN and network entities communicate with each other using MIH protocol messages. MIH protocol is employed to remotely send messages between separate MIHF entities. Once two MIHF entities need to communicate with each other, a transaction is initialised. An MIH transaction is a flow of messages with the same Transaction-ID submitted to, or received from, a particular MIHF ID. A specific MIH node cannot have more than one transaction pending for each direction with an MIH peer. According to classical MIH protocol, if the remote communication between entities is not reliable (frame loss probability ≥ 0.01), an acknowledgement service is required. Figure 2.3 shows the active state machines during a transaction. The MIH acknowledgement service uses two bits inside the MIH header: the ACK-Req bit is set by the source and the ACK-Rsp bit is set by destination. After sending an MIH protocol message with ACK-Req bit set, the source starts a retransmission timer.
and keeps a copy of it, while the timer is active. If the acknowledgement message is not received before the expiration of the timer, the source node retransmits the saved frame with the same Message-ID, with the same Transaction-ID and with ACK-Req bit set. Otherwise, if an acknowledgement was to be received before expiration of the timer or before any other retransmission attempt, the source ensures the correct reception of the message, resets the timer and deletes the saved copy of the packet. Moreover, if the MIH source receives an ACK for any of the previous transmission attempts then, the communication is classified as successful and it does not need to wait for any further acknowledgement. Retransmissions are done while ACK is received or the number of retransmissions reaches the maximum value.

On the destination side, frames received with the ACK-Req bit set cause the return of an MIH acknowledgement message with ACK-Rsp bit set in the header and with the same Message-ID and Transaction-ID. In particular, acknowledgements are MIH packets without payload. When an immediate response is waited by the MIH source, the receiver sends the corresponding MIH answer message with ACK-Rsp bit set. Then, the destination can also set the bit ACK-Req to ask the source to acknowledge the response message. If multiple messages are received, the sink only processes the first one. In this sense, all the duplicate frames are acknowledged.

Moreover, the MIH protocol does not provide direct support for congestion control and load management. Therefore, it is recommended to run the MIH protocol over congestion aware transport layers. In order to help prevent congestion, flow control mechanisms are implemented at the MIHF. A single rate limiter applies to all traffic (for all interfaces and message types). It applies to retransmissions, as well as new messages, although an implementation can choose to prioritize one over the other. When the rate limiter is in effect, MIH messages are queued until transmission is re-enabled, or an error condition is indicated back to local upper layer applications. The rate limiting mechanism is implementation specific, but it is recommended to use a token bucket limiter as described in IETF RFC 4443. When an MIHF suffers from overload, it drops requests from MIH requestors. Any reliable delivery function indicates a flow control back to the requestor, and an MIHF invokes flow control towards a specific requestor when overloaded with reliably delivered messages.
All MIH protocol messages have two identifiers: MIHF ID and Transaction-ID. The former uniquely identifies an MIHF entity to provide the services whilst the latter matches a request message with the correspondent response message or acknowledgement. An MIH protocol payload is constituted by a Source MIHF Identifier Type-Length-Value (TLV), a Destination MIHF Identifier TLV and an MIH service specific TLVs. Figure 2.4 depicts the fields of MIH protocol frame format and of its header. Table 2.1 describes the meaning of the different fields of the MIH protocol header format.

2.2 Burst Erasure Channel

As previously claimed in chapter 1, EC-MIH protocol outperforms MIH protocol during VHOs by deploying HSCNCs. In particular, we can take advantage of these novel erasure codes because VHOs can be modelled as burst erasure channels (this is explained in subsection 4.1.1). In this context, it is important to introduce the theory and the notation of burst erasure channels and burst erasure-correcting codes in order to aid the characterisation of subsequent system models in the next chapters.

The Burst Erasure Channel (BEC) is an erasure channel with memory that has applications in many contexts. For example, fading periods can be treated as erasure burst. Moreover, packet losses because of congestions and disconnections make BECs very attractive since real time or multicast applications can benefits from coding theory.

In general, channel models are of two kinds: generative and descriptive. A generative
Table 2.1: [2] MIH protocol header fields. Size is expressed in bits.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
<td>Current version of MIH protocol in use.</td>
</tr>
<tr>
<td>ACK-Req</td>
<td>1</td>
<td>Request an acknowledgment for the message.</td>
</tr>
<tr>
<td>ACK-Rsp</td>
<td>1</td>
<td>Answer to the request for and acknowledgment message.</td>
</tr>
<tr>
<td>Unauthenticated Information request (UIR)</td>
<td>1</td>
<td>It informs the MIH Information Service if the message is sent in pre-authentication/pre-association state so that, the length of the response message is limited.</td>
</tr>
<tr>
<td>More fragment (M)</td>
<td>1</td>
<td>It shows message is a fragment to be followed by another fragment.</td>
</tr>
<tr>
<td>Fragment number (FN)</td>
<td>7</td>
<td>It is used to represent the sequence number of the fragment (0-127).</td>
</tr>
<tr>
<td>Reserved (rsvd)</td>
<td>1</td>
<td>It is kept reserved and usually set to '0'.</td>
</tr>
<tr>
<td>MIH message ID</td>
<td>16</td>
<td>It is a combination of 3 different field: Service identifier (SID), operation code (Opcode) and action identifier (AID). The first identifies the different MIH services. The second is the kind of operation to be performed with respect to the SID. The third indicates the action to be done according to the SID.</td>
</tr>
<tr>
<td>Reserved2 (rsvd2)</td>
<td>4</td>
<td>Reserved field. Bits are usually set to '0'.</td>
</tr>
<tr>
<td>Transaction-ID</td>
<td>12</td>
<td>It is used to match requests, responses and acknowledgments.</td>
</tr>
<tr>
<td>Variable payload length</td>
<td>16</td>
<td>Total length of variable payload of respective MIH protocol frame.</td>
</tr>
</tbody>
</table>
Figure 2.5: Gilbert’s model for burst error channels is a two-state Markov model. $S_1$ and $S_2$ are respectively the 'Good' and 'Bad' state and the probabilities $p_{ij}$ are the transition probabilities.

model \cite{35} is a Markov chain consisting of a finite or infinite number of states with defined transition probabilities. The transitions among the states produce a sequence. For example, this sequence can be mapped to a corresponding digit sequence. The use of a channel model allows to simulate real channel behaviours. The model is suitably parameterized using real data collected from the channel. On the other hand, a descriptive model uses data analysis of the error sequences to generate a statistical model of the channel. In this case, it is important to consider the trade-off between accuracy of the model and complexity. Derivation of the characteristics of a channel from its brute force direct processing is usually difficult and requires a lot of computing time. So, the design of channel models has to achieve accuracy, but with acceptable complexity.

2.2.1 Gilbert’s Channel

Gilbert’s model \cite{36} is a useful theoretic representation of a channel with memory, which can describe BECs’ behaviours as well. Gilbert’s model is a generative model. The classical form of a Gilbert’s channel is a finite-state Markov model with two states (see figure 2.5). The two state $S_1$ and $S_2$ are respectively the 'Good' and 'Bad' state. An error sequence is generated by the Markov chain. In state $S_1$, there are no errors due to erasures while in state $S_2$, erasures occur with some probability. After producing the error symbol at time $t$, the chain makes a transition to prepare the error symbol at time $t + 1$. The transition probabilities are $p_{ij}$, whose values have to guarantee the
persistence of the states \( S_1 \) and \( S_2 \) in order to simulate the burst condition.

A detailed description of Gilbert’s channel can be found in [35, 36]. It is fundamental to notice that the accuracy of this model mainly depends on the choice of the distributions assigned to \( p_{ij} \): this represents the principal task when Gilbert’s channel model is applied.

2.3 'Classical’ Error-Correcting Codes

This section introduces the basic notation and concepts of coding theory [37, 38, 39, 40] that are important to understand the work of this thesis and its results.

Let \( \mathbb{F}_q \) be an extension field of size \( q \), where \( q = \chi^m \) is power of the prime number \( \chi \). Then, let \( \mathbb{F}_q^n \) be the \( n \)-dimensional vector space over \( \mathbb{F}_q \). So, a \( q \)-ary linear code \( C(n,k) \) can be defined as a \( k \)-dimensional linear subspace of \( \mathbb{F}_q^n \). The vectors in \( C \) over the alphabet \( \mathbb{F}_q \) are the codewords of the code. As an example, the codes defined over the field/alphabet \( \mathbb{F}_2 \) are called binary codes.

The most common ways to introduce a linear code are by using a generator matrix and a parity-check matrix. A generator matrix \( \mathbf{G} \) for a linear code \( C \) is a \( k \times n \) matrix whose rows form a basis for \( C \). Side by side, a parity-check matrix \( \mathbf{H} \) for \( C \) is a \((n - k) \times n \) matrix. This matrix represents the generator matrix for \( C^\perp \), the dual or orthogonal linear code of \( C \).

An important invariant of a code is the minimum distance between codewords. Let Hamming distance \( d(x,y) \) between two vectors \( x, y \in \mathbb{F}_q^n \), be the number of coordinates in which \( x \) and \( y \) differ. Distance is a metric on the linear space \( \mathbb{F}_q^n \). Then, the minimum distance of a code is \( d_{\text{min}} = \min \{ d(x,y) : x,y \in C, x \neq y \} \): it is the smallest distance between distinct codewords and is important in determining the error-correcting capabilities of a code \( C \). Next, the Hamming weight of a vector \( x \in \mathbb{F}_q^n \) is the number \( wt(x) \) of its nonzero coordinates. Clearly, \( d(x,y) = wt(x - y) \). Thus, if \( C \) is a linear code, the minimum distance is the same as the minimum weight of a nonzero \( \dagger \)

\[ \text{The dual or orthogonal linear code of } C \text{ is defined by } C^\perp = \{ v \in \mathbb{F}_q^n | u \cdot v = 0, \forall u \in C \}. \]
codeword. If the minimum weight of a \((n, k)\) code is known, then the code is referred to as \((n, k, d_{\text{min}})\) code.

After having introduced the previous general definitions, it is necessary to discuss some other properties of codes, such as the Singleton bound. This represents a bound on the size of an arbitrary block code, and it is relevant for some important definitions. First, let \((n, M, d_{\text{min}})\) code be a code of length \(n\) consisting of \(M\) codewords with minimum distance \(d_{\text{min}}\). \(A_q(n, d_{\text{min}})\) is defined as the maximum number of possible codewords in the \(q\)-ary block code of length \(n\) and minimum distance \(d_{\text{min}}\). Hence, the Singleton bound \[41\] states that

\[
A_q(n, d_{\text{min}}) \leq q^{n-d_{\text{min}}+1}
\] (2.1)

By considering a linear code \(C(n, k)\), this bound becomes \(k \leq n-d_{\text{min}}+1\). A linear code, which has \(d_{\text{min}} = n - k + 1\), is called Maximum Distance Separable (MDS) code. The main property of this family of linear codes is that it maximises the possible distance between the codewords. A linear code has to satisfy the condition \(d_{\text{min}} \leq n - k + 1\) because of the Singleton bound.

Finally, the code \(C(n, k)\) is characterised by the parameter \[37\]

\[
R = \frac{k}{n}
\] (2.2)

which is called the rate, and it is a value always less than one since the code introduces redundancy. Error-correcting codes can be classified into two main families according to the properties of their rate: rate and rateless. While a rate code has a fixed designed rate that if changed, alters the characteristics of the code itself, a rateless code can generate a number of encoded data that is potentially limitless. In particular, codewords can be generated on the fly, as few or as many as needed. On the other side, the decoder can recover an exact copy of the data from any received set of codewords.

2.3.1 Reed-Solomon Codes

There are different ways to define RS codes. A frequent approach in the literature is the one that considers RS codes as cyclic codes.
2.3. ‘Classical’ Error-Correcting Codes

Figure 2.6: Folding of a RS code with parameter $\theta$. The arrows clarify the correspondence between the RS code and its respective FRS code.

RS codes [7] are special cases of Bose-Chaudhuri-Hocquenghem (BCH) codes. In particular, an RS code over $\mathbb{F}_q$ is a BCH code of length $n = q - 1$. Let $\gamma$ be a primitive element\(^2\) of $\mathbb{F}_q$ and let $k$ be an integer with $0 \leq k \leq n = q - 1$. Then

$$C = \{(f(1), f(\gamma), f(\gamma^2), \ldots, f(\gamma^{q-2})) \mid f \in \mathfrak{P}_k\} \quad (2.3)$$

is the narrow sense RS code $C(n, k, d_{\text{min}})$ over $\mathbb{F}_q$, with $d_{\text{min}} = n - k + 1$ since it is an MDS code. If $k \geq 0$, $\mathfrak{P}_k$ represents the set of polynomials of degree less than $k$ in $\mathbb{F}_q[x]$, including the zero polynomial. In particular, the polynomials in $\mathfrak{P}_k$ are in the form $f(x) = u_0 + u_1 x + \ldots + u_{k-1} x^{k-1}$, and have coefficients $(u_0, u_1, \ldots, u_{k-1})$, which are the $k$ information symbols. Furthermore, equation (2.3) states that the set of codewords is obtained by evaluating the polynomial $f(x)$ in $\gamma$ and its powers until $\gamma^{q-2}$.

Folded Reed-Solomon codes [42, 43] are codes $\text{FRS}_q(\theta)(n, k)$ over alphabet $\mathbb{F}_q^\theta$. They are encoded by a polynomial of degree at most $k - 1$ as depicted in figure 2.6. Moreover, figure 2.6 also shows the one-to-one correspondence between RS codes and FRS codes. The block length of $\text{FRS}_q(\theta)(n, k)$ is $\ell = n/\theta$ and its rate is equal to the original unfolded RS code (obtained by choosing $\theta = 1$). The difference is that the FRS code is defined over a larger alphabet, of size $q^\theta$.

Normally, the decoder of RS codes uses the Berlekamp-Massey algorithm [40, Chapter

---

\(^2\)A primitive element of a field $\mathbb{F}_q$ is an element $\alpha$ of order $(q - 1)$ in $\mathbb{F}_q$, whose power sequence $\alpha^1, \alpha^2, \ldots, \alpha^{q-1} = 1 = \alpha^0$ contains all the nonzero elements of $\mathbb{F}_q$. 


2.3. ‘Classical’ Error-Correcting Codes

Figure 2.7: Simple example of a parity-check matrix $H$ and its respective Tanner graph. The labels C1, C2, C3 and C4 are the parity-check equations (check nodes), and $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, $x_6$, $x_7$ and $x_8$ are the code symbols (variable nodes). The edges identify the 1s in $H$.

6]. This algorithm is valid in any field but it has some problems of numerical precision in real and complex fields. A recent efficient algorithm for FRS codes is described in [43].

2.3.2 Low-Density Parity-Check Codes

Low-density parity-check codes are binary block codes with very low complexity of decoding. The main design requirement is to fix the number of ones in the rows and the columns of the parity matrix $H$. Moreover, LDPC codes have very small number of ones compared to the size of the matrix, so $H$ results to be sparse. In particular, the rows of $H$ represent the parity check equations, and the columns are the digits in the codeword.

Figure 2.7 represents a simple example of an LDPC code. The labels C1, C2, C3 and C4 identify the parity check equations, while $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, $x_6$, $x_7$ and $x_8$ are the code symbols. Matrix $H$ can be represented as a Tanner graph, in which the
connections between the variable nodes (referred to code symbols) and the check nodes are the 1s in the matrix. A common method to decode LDPC codes is called iterative message-passing. An iterative message-passing algorithm consists in all variable nodes and all check nodes iteratively passing messages in parallel along the adjacent edges of the graph.

Given these premises, a stopping set \( S \) is a subset of the set of variable nodes, such that every check node connected to \( S \), is connected to \( S \) at least twice. If all of the bits in a stopping set are erased none of them can be corrected using message-passing decoding: so a stopping set distribution of an LDPC code determines when the message passing decoder can fail. Especially, in a BEC the location of the stopping set bits within a codeword is an important parameter to determine the performance of the message-passing decoder.

The LDPC codes were generally proposed for memoryless channels. However, reference [8] presented optimal LDPC codes for single and multiple burst erasure correction. That article showed a deterministic construction method to design \( H \) capable to go close to optimal burst erasure correction: this method is based on superposition.

The superposition method starts with a \( \Phi \times N \) base matrix \( H_{\text{base}} \). Next, the entries in \( H_{\text{base}} \) are replaced by \( v \times v \) binary matrices, called superposition matrices, to create an \( \phi \times n \) LDPC code parity-check matrix \( H \), with \( \phi = \Phi v \) and \( n = N v \). Each zero entry in \( H_{\text{base}} \) is replaced by \( v \times v \) zero matrix, and each non-zero entry in \( H_{\text{base}} \) by a \( v \times v \) circulant or permutation superposition matrix. A \( v \times v \) circulant matrix \( F \) is defined by the polynomial \( f(x) = a_1 + a_2x + a_3x^2 + \cdots + a_vx^{v-1} \), where coefficient \( a_i \) is the entry in the \( i \)-th row of the first column of \( F \) and can have a value of '0' or '1'. Figure 2.8 depicts a simple example of how superposition method works. The base matrix has \( \Phi = 1 \) and \( N = 4 \), while the circulant and zero matrices have \( v = 3 \).

The production of highly efficient burst erasure-correcting codes is obtained with the optimal base matrices

\[
H_{\text{base}} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \quad (2.4)
\]
2.3. ‘Classical’ Error-Correcting Codes

Figure 2.8: Simple example of superposition method to build a parity-check matrix for an LDPC code. The starting point is a matrix $H_{\text{base}}$ of size $\Phi \times N$, with $\Phi = 1$ and $N = 4$. The entries with 1s are substituted by $v \times v$ circulant matrix $F$, and the 0s are substituted by $v \times v$ zero matrices. In this particular example $v = 3$, so the parity matrix $H$ becomes a $3 \times 12$ matrix.

![Diagram of H_base and H matrices](image)

and

$$
H_{\text{base}} = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & \cdots & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0
\end{bmatrix}
$$

Matrix (2.4) provides an MDS base matrix, while matrix (2.5) has the drawback of producing codes with low rate.

If $H_{\text{base}}$ has structure (2.4), there are two possible code constructions. First, Construction 1 replaces $i$-th entry of $H_{\text{base}}$ with the $v \times v$ circulant $F_i$, where

$$
f_i(x) = 1 + x^{b_i} \\
b_i = \left\lfloor \frac{v}{2} \right\rfloor - i
$$

Next, Construction 2 replaces $i$-th entry of $H_{\text{base}}$ with the $v \times v$ circulant $F_i$ ($v > 8N$), where

$$
f_i(x) = 1 + x^{b_i} + x^{c_i} \\
b_i = 2i \\
c_i = \left\lceil \frac{3v}{8} \right\rceil + i
$$

Both Construction 1 and Construction 2 provide 4-cycle free parity-check matrices.

If $H_{\text{base}}$ has structure (2.5), there are other two possible code constructions. Construction 3 builds a $3 \times 3\tilde{p}$ base matrix from the concatenation of $\tilde{p}$ copies of $3 \times 3$ base...
2.3. ‘Classical’ Error-Correcting Codes

matrices from (2.5). Then the parity-check matrix $H$ is produced using superposition. The last nonzero entry in the columns of $H_{\text{base}}$ corresponding to the $i$-th copy of (2.5), $i \in \{1, \ldots, \tilde{p}\}$, is replaced with $I_{\nu}^{(i+1)}$. Every other non-zero entry of $H_{\text{base}}$ is replaced by $I_{\nu}$ and every zero entry by zero matrix. Next, Construction 4 builds the $N \times \tilde{p}N$ base matrix from the concatenation of $\tilde{p}$ copies of the $N \times N$ base matrices from (2.5). Then, parity-check matrix is done using superposition where every zero entry in the $H_{\text{base}}$ is replaced by zero matrix, the first non-zero entry of each column of $H_{\text{base}}$ is replaced by the $I_{\nu}$, and the second non-zero entry of each column of $H_{\text{base}}$ is replaced by $I_{\nu}^{(l_i)}$ with the orders $l_i$ chosen so that the sum of the shift orders in any adjacent set of $N$ columns is non-zero mod $N$.

2.3.3 Fountain Codes

Fountain codes are effective rateless codes that can provide reliable transmissions. Formally, for a given vector $u$ of $k$ source symbols in $\mathbb{F}_{q_1}$, a fountain encoder produces a potentially limitless stream of encoded symbols $x_1, x_2, x_3, \ldots$ in $\mathbb{F}_{q_2}$, with $q_1$ not necessarily equal to $q_2$. In general, the size of the finite field is dictated by the requirements. For simplicity let’s discuss a code that operates on binary symbols. Fountain codes are governed by a probability distribution on a vector space $\mathbb{F}_2^k$. In order to generate the encoding symbols, the encoder samples the probability distribution to obtain a vector $(v_1, v_2, \ldots, v_k) \in \mathbb{F}_2^k$ and then, it calculates $x_j = \sum_i v_i u_i$. Since the samplings of the fountain encoder are independent from encoded symbol to encoded symbol, these symbols have a uniformity property that ensures the fountain property of the code.

The average computational cost for the generation of an encoded symbol is the average weight of the vector $(v_1, v_2, \ldots, v_k)$ (when sampled from the probability distribution) multiplied by the computational cost of adding two symbols together. So, it is important to keep the average weight as small as possible. In parallel, the characteristics of the probability distribution have high impact on the decoder performance. Another main property of a fountain code is its overhead, that is the quantity of received symbols over the first $k$, required to decode with high probability the source symbols. The overhead is $\phi$ if $k + \phi$ encoded symbols are used to attempt decoding. Finally, the
2.3. 'Classical' Error-Correcting Codes

Figure 2.9: Simple example of an LT code. The symbols $u_i$ are the source symbols, while symbols $x_i$ are the encoded symbols. The bipartite graph shows the connections between source symbols and encoded symbols, represented by the edges. The number of edges connected to a node, referred to an encoded symbol, is called the degree.

decoding failure probability is independent of the pattern of received symbols but is only a function of the overhead $\phi$.

Luby Transform Codes

Given a coding scheme, decoding is usually the process that requires higher complexity. An LT code uses the Belief-Propagation (BP) decoder. Figure 2.9 depicts an example of how LT encoder works. According to a distribution that provides the degree for each encoded symbol, the source symbols are added up to output the stream of encoded symbols. The BP decoder works on the bipartite graph in order to decode the received symbols. Especially, this decoding algorithm works as follow. First, it looks for an encoded symbol of degree one ($x_3$ in figure 2.9) and if none exists, it returns a decoding failure. On the other hand, a symbol of degree one directly recovers the value of its neighbouring source symbol. Once this symbol is recovered, its value is added to the ones of all its neighbouring encoding symbols. Finally, the connections of this symbol
with other encoding symbols are removed. These steps are repeated till the decoding process is completed.

Some problems can arise at the decoder: in fact, there may not be encoded symbols of degree one at some intermediate step of the decoding to lead either to a decoding failure or to a high overhead. The main way to reduce or to avoid these issues is to carefully design the probability distribution of the fountain code. The property of LT codes is the capability to provide an efficient distribution for the BP process, called Robust Soliton distribution. When the probability distribution \( \Omega \) is fixed, probability values are assigned to the values \( \{1, 2, \ldots, k\} \). \( \Omega \) induces a probability distribution \( \Theta \) on \( \mathbb{F}_2^k \), which assigns a probability \( \Omega_{wt}/(\binom{k}{wt}) \) to a vector of Hamming weights \( wt \). A sample \( wt \) from \( \Omega \) is used to obtain an integer \( wt \) and then a vector of weight \( wt \) in \( \mathbb{F}_2^k \) is uniformly and at random sampled from \( \Theta \). So the pair \( (k, \Omega) \) fully characterises the LT code.

An LT code starts with an LT encoder receiving a stream of \( L \) source bits. Next, it partitions them into \( k = L/m \) input symbols over \( \mathbb{F}_q \): The encoding symbols are created according to the distribution \( \Omega \) (Robust Soliton distribution) \[9\], which is the sum between the distribution

\[
\tau(i) = \begin{cases} 
\frac{s}{ik} & \text{for } i = 1, \ldots, k/s - 1 \\
\frac{s \log(s/\delta)}{k} & \text{for } i = k/s \\
0 & \text{for } i = k/s + 1, \ldots, k 
\end{cases} \tag{2.8}
\]

and the ideal Soliton distribution

\[
\rho(1) = \frac{1}{k} \\
\rho(i) = \frac{1}{i(i-1)} \quad \text{for } i = 2, \ldots, k \tag{2.9}
\]

Hence, after normalising, the result becomes

\[
\beta = \sum_{i=1}^{k} \rho(i) + \tau(i) \\
\mu(i) = (\rho(i) + \tau(i))/\beta \quad \text{for } i = 1, \ldots, k \tag{2.10}
\]

The parameter \( \delta \) represents the upper bound on the failure probability at the decoder, given a set of \( n \) encoded symbols. Next, the variable \( s \) is defined as \( c\sqrt{k} \log \left( \frac{k}{\delta} \right) \), where \( c > 0 \) is a constant.
An important parameter of LT codes is the average degree $D$ of an encoding symbol. Each encoding symbol is obtained by adding up a random number of source symbols, called degree. Then, the number of source symbols used on average to create an encoded symbol is the average degree of the code. The value of $D$ mainly depends on $k$ and $\delta$ according to a factor $\log(k/\delta)$ [9]. This is the number of symbol operations on average the decoding of an encoded symbol requires. Thus, the aim is to keep $D$ as small as possible. The computational complexity of BP decoding is linear in the average degree multiplied by the size of the source block.

Figure 2.10 shows how the parameter $\delta$ affects the variation of the Robust Soliton distribution, given a fixed number of source symbols $k$. In particular, figure 2.10(a) and 2.10(b) respectively show the variation of the distribution for $\delta \leq 0.1$ and for $\delta > 0.1$. In parallel, figure 2.11 is significant to understand how changing $c$ parameter influences the Robust Soliton distribution of an LT code.

Finally, figure 2.12 and figure 2.13 clarify how different Robust Soliton distributions vary the distribution of the degrees of the encoded symbols, and so the average degree $D$. Especially, the degree of each encoded symbol for the two selected distributions are represented by the bars in figure 2.12(b) and figure 2.13(b). The former distribution has two peaks with higher probability for degree 2 and 11 (figure 2.12(a)), while the latter has higher values for degree lower than 6 (figure 2.13(a)). Thus, the first distribution provides encoded symbols with degree higher (figure 2.12(b)) than the second one (figure 2.13(b)).

**Raptor Codes**

Raptor codes are an extension of LT codes. In particular, Raptor codes use a pre-coding stage to encode source symbols and then pass these pre-encoded symbols to LT encoder. This serial concatenation permits to significantly reduce the probability of the LT code to have a constant fraction of source symbols not contributing to the values of any of collected encoded symbols. The pre-coded symbols are called *intermediate symbols*. Next, a suitable LT code with constant average degree is applied to the intermediate symbols to produce the encoded symbols. Once the LT decoder ends its
2.3. ‘Classical’ Error-Correcting Codes

Figure 2.10: Simulations to show how parameter $\delta$ affects the Robust Soliton distribution. The simulations consider an LT code of $R = 1/2$, with $n = 100$, $k = 50$ and $c = 0.1$. The snapshots consider the $k$ between 0 and 20 since the values of the distribution for $k$ greater than 20 are negligible (a) Robust Soliton distribution with $\delta \leq 0.1$ (b) Robust Soliton distribution with $\delta > 0.1$. 
2.3. ‘Classical’ Error-Correcting Codes

Figure 2.11: Simulations to show how parameter $c$ affects the Robust Soliton distribution. The simulations consider an LT code of $R = 1/2$, with $n = 100$, $k = 50$ and $\delta = 0.1$. The snapshot considers the $k$ between 0 and 20 since the values of the distribution for $k$ greater than 20 are negligible.
Figure 2.12: Characterisation of the Robust Soliton distribution of an LT code of $R = 1/2$, with $n = 100$, $k = 50$, $\delta = 0.1$, $c = 0.1$ and with maximum weight per encoded symbol equal to 22 (a) Graph of the Robust Soliton distribution depending on degree (b) Distribution of the weight of each encoded symbol. The horizontal axes considers all the 100 encoded symbols.
2.3. ‘Classical’ Error-Correcting Codes

Figure 2.13: Characterisation of the Robust Soliton distribution of an LT code of $R = 1/2$, with $n = 100$, $k = 50$, $\delta = 0.0001$, $c = 0.06$ and with maximum weight per encoded symbol equal to 17 (a) Graph of the Robust Soliton distribution depending on degree (b) Distribution of the weight of each encoded symbol. The horizontal axes considers all the 100 encoded symbols.
operations, a small fraction of the intermediate symbols will still be unrecovered. If the pre-code is appropriately chosen, then this set can be recovered using an erasure decoding algorithm for the pre-code.

More advanced Raptor codes provide lower decoding failures than LT codes, given same overhead. Raptor codes achieve linear time encoding and decoding performance based on a simple idea: an appropriate linear block code $C(n, k)$ is used to encode the vector $u_1, ..., u_k$ of source symbols to generate redundant symbols $x_1, ..., x_n$, where $n$ is a small fraction of $k$. The concatenation $u_1, ..., u_k, x_1, ..., x_n$ of the source symbols and the redundant symbols is called the intermediate block. There are $n$ constraints that define the relationship between the source symbols and the encoding symbols of the intermediate block, and these constraints can be viewed as symbols, hereafter called constraint symbols. The value of each constraint symbol is zero, i.e. the constraint symbol constrains the sum of its neighbouring intermediate symbols to be equal to zero.

The intuitive advantage of pre-coding is that the redundancy amongst the intermediate symbols allows recovery of all the intermediate symbols using the decoder for $C(n, k)$ if most of the intermediate symbols are known. Some codes, which are suggested for pre-coding are [11]: LDPC codes, Tornado codes and Repeat-Accumulate codes.

### 2.4 Network Coding and Network Error-Correcting Codes

As previously specified in chapter 1, there are two main ways to define a network code, one mainly based on graph theory and another one fully based on algebra and matrices. Figure 2.14 shows a comparison between these two methods. On the left, a linear code multicast defined as a local and global encoding mapping: $K$ represents the local encoding kernel at a node and $f$ is the global encoding kernel at an edge. Local and global encoding kernels describe the 2-dimensional linear network code for this multicast scenario with a source and two sinks. The ingoing channels of the source are called imaginary channels because they have no originating nodes. On the right, a description of the linear network code through the algebraic framework. $X$ is the source random process, $Y$ is random process at an edge and $Z$ is the random process.
Figure 2.14: On the left, a linear code multicast defined as a local and global encoding mapping, with $K$ (the local encoding kernel at a node) and $f$ (the global encoding kernel at an edge). On the right, a description of the linear network code through the algebraic framework. $X$ is the source random process, $Y$ is random process at an edge and $Z$ is the random process collected by a sink. Next, $\alpha$, $\beta$ and $\varepsilon$ are constant coefficients in $\mathbb{F}_2$. Matrices $A$ and $B$ contain the coefficients, while matrix $F$ is the adjacency matrix of the directed labelled line graph. Finally, $M$ is the transfer matrix of the network [41].
collected by a sink. Next, $\alpha$, $\beta$ and $\varepsilon$ are constant coefficients in $\mathbb{F}_2$. These coefficients represent the elements of matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{F}$. In particular, matrix $\mathbf{F}$ is the adjacency matrix of the directed labelled line graph, obtained from the graph of the network (see the graph on the extreme right.). Finally, it is possible to see how the transfer matrix $\mathbf{M}$ is calculated by using the coefficients $\alpha$, $\beta$ and $\varepsilon$ of the linear combinations.

Random linear network coding is fully based on the algebraic definition of a network code. This random network code is a distributed network coding scheme, which is useful in noncoherent network models\textsuperscript{3}. In particular, the noncoherent network model fits well in the requirements of most real applications. The encoded packets are generated by RLNC encoder as

$$\mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{A} \cdot \mathbf{U} \end{bmatrix}$$  \hspace{1cm} (2.11)

where $\mathbf{U}$ is the matrix of source packets, $\mathbf{A}$ is the matrix of random coefficients and $\mathbf{X}$ is the matrix of encoded packets. All the matrices $\mathbf{U}$, $\mathbf{A}$ and $\mathbf{X}$ have elements in $\mathbb{F}_q$. After that, even if an intermediate node performs re-encoding on these linear combinations, the effect can still be represented by (2.11), with $\mathbf{A}$ being replaced by the overall transfer matrix. Upon receiving the packets, the receiver needs to calculate $\mathbf{A}^{-1}$ using Gauss-Jordan elimination, and to apply the corresponding linear operations on the received packets to obtain the original message. In matrix form, the decoding process is represented as

$$\begin{bmatrix} \mathbf{A}^{-1} \cdot \mathbf{A} & \mathbf{A}^{-1} \cdot \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{U} \end{bmatrix}$$  \hspace{1cm} (2.12)

where $\mathbf{I}$ is the identity matrix. So, the decoder needs to receive as many linear independent rows of $\mathbf{X}$ as the number of rows of $\mathbf{U}$ in order to be able to decode.

Network error correction coding is a generalisation of ‘classical’ error correction coding. While the latter achieves error control by adding redundancy in time domain, the former introduces redundancy in space domain. Reference [21] and [22] extended the bounds in classical coding theory (see section 2.3) to NEC theory. Especially, the Singleton bound is of great importance for coding design. Let $\mathcal{V}$ be the set of nodes of a network

\textsuperscript{3}A network is called coherent if network characteristics are known to sources and sinks, and it is called noncoherent if they are unknown.
and let \( \{V_1, V_2\} \) be a partition of \( V \). The cut\(^4\) for this partition is \( \text{cut}(V_1, V_2) = \{e \in E : \text{tail}(e) \in V_1, \text{head}(e) \in V_2 \} \). According to Max-Flow Min-Cut theorem [44, 45] in a network, the value of the maximum flow is equal to the capacity \( C_\lambda \) of the minimum cut. Then, if there exists an \( \omega \)-dimensional error-correcting code that can correct up to \( \iota_\lambda \) errors for every sink \( \lambda \in \Lambda \), the refined Singleton bound [46] becomes

\[
\omega + 2\iota_\lambda \leq C_\lambda \tag{2.13}
\]

or equivalently

\[
2\iota_\lambda \leq \phi_\lambda \tag{2.14}
\]

Next, as in 'classical' coding theory, MDS network error-correcting codes are the ones that satisfy the equality of the refined Singleton bound.

If the network code has to be designed independently of the network topology, a RLNC solution is required. Nevertheless, the price to pay while using RLNC is that the decoding matrix becomes random: hence, with a certain probability it could not be full rank and the decoding fails. This decoding failure probability \( P_e \) [47, 48] is characterised by the upper bound

\[
P_e^\lambda \leq \frac{\phi_\lambda + |J| + 1}{(q - 1)^{\phi_\lambda + 1}} \tag{2.15}
\]

where \( \phi_\lambda \) is the redundancy for sink \( \lambda \), and \( J = V - \Sigma - \Lambda \). Especially, \( \Sigma \) is the set of source nodes and \( \Lambda \) the set of sink nodes. Hence, \( J \) is the set of nodes of the network that are neither source nor sink nodes. The bound in relation (2.15) demonstrates how the size of the network, the redundancy of the code and the size of the finite field, affects the decoding error probability.

Because of that, for RLNC the minimum distance of the code is a random variable taking values in the range \( 1 \leq D_{\min} < \phi_\lambda + 1 \). The probability mass function [48, 46] of the minimum distance of a random linear NEC at a sink node \( \lambda \in \Lambda \) mainly depends on the size of the network \( J \), the redundancy of the code at a sink \( \phi_\lambda \), and the inverse of the size of the finite field \( q \). The minimum distance of RLNC is denoted by a capital letter because it is a random variable while \( d_{\min} \) is used for deterministic codes.

\(^4\)A cut of a \( \sigma-\lambda \) network is a set of edges such that if these edges are removed from the graph, the source node \( \sigma \) and the sink node \( \lambda \in \Lambda \) are disconnected. The capacity of a cut is the sum of the capacities of the edges in the cut.
2.5 Summary

This chapter has discussed the existing techniques to perform VHOs between the three main technologies IEEE 802.21, IEEE 802.16 and 3GPP. Moreover, it has explained the architecture, the functionalities and the protocol developed in the standard IEEE 802.21: this solution provides technology-agnostic VHOs.

Next, the explanation has moved to the theory of burst erasure channels and burst erasure-correcting codes, together with the introduction of the mathematical notation that is used in the rest of the thesis. Furthermore, the chapter underlined and described the challenges related to the design of effective erasure codes and NEC schemes.

RS codes are very powerful erasure codes, but they can achieve high complexity depending on the selected $q$. LDPC codes are efficient and low complexity codes, but they require a careful design of their parity-check matrix $H$: this matrix fully determines the erasure-correcting capabilities of these codes. Fountain codes are novel coding schemes, very efficient and very effective for erasure correction. However, they rely on a generator matrix that is sampled from a random distribution. Hence, the structure of the distribution affects the performance of the codes. Moreover, Raptor codes also require special attention in the choice and design of the pre-coding scheme. Finally, RLNC is a noncoherent NEC that introduces redundancy in space and not in time. It provides several benefits in terms of erasure correction and throughput. However, RLNC can have some important drawbacks: the choice of $q$ influences the decoding error probability and the complexity of the encoding/decoding operations, the minimum distance becomes a random variable, and the transmission of the random coefficients can add significant overhead to the communication.

The thesis will address all the above issues by carefully designing efficient HSCNCs that aim to optimise all the aforementioned parameters.
Chapter 3

Hybrid Serial Concatenated Network Codes for Burst Erasure Channels

Chapter Outline: This chapter describes the structure and the performance of HSCNCs for burst erasure channels. This novel coding scheme is obtained by serially concatenating an erasure code as outer code and systematic RLNC as inner code. The main objective is to outperform RLNC in terms of erasure correction capabilities and required redundancy. A related design requirement is not to significantly increase the encoding/decoding delay. We first provide an introduction about the existing ways to outperform RLNC performance. Moreover, we show erasure-correcting capabilities of RLNC. Next, we analyse the decoding error probability and the encoding/decoding latency of HSCNCs with outer codes such as LT codes, RS codes, LDPC codes and Raptor codes, and systematic RLNC performing operations over \( \mathbb{F}_4 \) and \( \mathbb{F}_{256} \). In parallel, we also provide theoretical work to analytically justify the obtained improvement by using HSCNCs instead of RLNC alone. With the help of MATLAB simulations, we test different size of the finite field of RLNC in order to evaluate the impact of the decoding failure probability of RLNC on the decoding error probability of HSCNCs. Moreover, we study and define the best robust Soliton distribution for LT outer codes to optimise the performance of HSCNCs. Finally, we propose a new type of Raptor codes,
3.1 Introduction

Figure 3.1 depicts a basic line network, in which a source communicates with a sink via a link, which experiences burst erasures with some probability. If the source encodes the information packets with RLNC, it needs to send at least $D_{\text{min}} - 1$ redundant packets to guarantee a reliable communication. This condition is necessary to allow the sink to decode the received packets, hence to receive a full rank matrix $X$ to perform Gaussian elimination. Figure 3.2 shows the output of a RLNC encoder for erasure correction: the encoded matrix $X$ contains $k + \phi$ linear combinations of source packets, and has rank($X$) = $k$. If the erased packets are $\geq \phi + 1$, then rank($X$) $< k$ and the decoding error probability becomes 1. Since in all the rest of the chapter the transmissions will involve just one sink ($\lambda = 1$), we will simplify the notation by writing $\phi$ instead of $\phi\lambda$.

Regarding RLNC decoding performance, Trullols-Cruces et al. [49] have obtained the analytical expression of the exact decoding probability of RLNC. Let $b_e$ be the burst erasure probability of the channel, and let $n = k + \phi$ be the encoded output. Then, their result can be written as

$$P_{RLNC} = \sum_{j=k}^{n} \binom{n}{j} (1 - b_e)^j b_e^{n-j} p_{ns}(j, k)$$

where

$$p_{ns}(j, k) = \begin{cases} 0, & \text{if rank}(\tilde{X}) < k \\ \prod_{j=0}^{k-1} \left(1 - \frac{1}{q^j}\right), & \text{if rank}(\tilde{X}) > k \end{cases}$$

Figure 3.1: Line network, in which a source communicates with a sink via a burst erasure channel.
3.1. Introduction

Figure 3.2: Output of a RLNC encoder that provides erasure correction. The matrix $X$ is the encoded matrix, which contains $k$ encoded linearly independent packets and $\phi_\lambda$ redundant encoded packets.

\[ \text{rank}(X) = k \]

where $P_{RLNC}$ is the probability of decoding received packets, and $\tilde{X}$ is the matrix got by the decoder. Thus, the decoding error probability becomes $P_{e_{RLNC}} = 1 - P_{RLNC}$. Figure 3.3 depicts the decoding error probability of RLNC in presence of losses, for coding rates $R = 0.76$ and $R = 0.86$, in $\mathbb{F}_4$ and $\mathbb{F}_{256}$.

In order to handle erasures more efficiently and to enhance the performance of RLNC, two main solutions have been proposed in the literature. The first deploys systematic RLNC [50]. The second develops novel coding schemes, called Batched Sparse (BATS) codes [51, 52, 53].

Systematic RLNC is a systematic network coding scheme, which reduces the complexity of encoding/decoding operations of RLNC, while also reducing its decoding error probability. In fact, since only redundant packets are network coded packets, when the received rank becomes less than $\text{rank}(X)$, the decoding error probability can be still $< 1$.

On the other side, BATS codes consist in a sort of serial concatenation of fountain codes (either LT codes or Raptor codes), and RLNC. This concatenated scheme divides the source packets into sets (batches), that are encoded via fountain codes with differ-
Figure 3.3: Representation of decoding error probability of RLNC in $\mathbb{F}_4$ and $\mathbb{F}_{256}$. The initialisation sets $\text{rank}(X) = 32$, and coding rates $R = 0.76$ and $R = 0.86$.

ent generator matrices $G$. Then, only the packets within the same batch are linearly combined with RLNC. BATS codes are very efficient codes both in terms of encoding/decoding complexity and in terms of decoding error probability. Nevertheless, if the full rank of $X$ is not achieved in each batch at the receiver because of erasures, the Gaussian elimination for each batch cannot be performed, and the fountain code has no correcting power.

Table 3.1 summarises the encoding and decoding complexity of RLNC, BATS codes and systematic RLNC. Their comparison shows that BATS codes are the lowest complex codes, and that systematic RLNC significantly reduces encoding/decoding complexity of RLNC. While for RLNC the complexity depends on the whole size of the data block ($k$ packets), the complexity of systematic RLNC only depends on the size of the redundancy ($\phi$ packets).

This chapter describes and studies HSCNCs. These new coding schemes efficiently merge some of the characteristics of both systematic RLNC and BATS codes. HSCNCs extend the idea of serial concatenation proposed for BATS codes: in fact the outer
Table 3.1: Encoding and decoding complexity of RLNC, BATS codes and systematic RLNC.

<table>
<thead>
<tr>
<th>Coding scheme</th>
<th>Encoding</th>
<th>Decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLNC [52]</td>
<td>$O(lk^2)$</td>
<td>$O(k^3 + lk^2)$</td>
</tr>
<tr>
<td>BATS [52]</td>
<td>$O(lk(batchsize))$</td>
<td>$O(k(batchsize)^2 + lk(batchsize))$</td>
</tr>
<tr>
<td>Systematic RLNC [50]</td>
<td>$O(l\phi_1^3)$</td>
<td>$O(k^3b_1^2)$</td>
</tr>
</tbody>
</table>

Figure 3.4: Encoder structure of a HSCNC.

A code can be any erasure code (not only a fountain code). On the other hand, the main difference is that HSCNCs do not map source packets into batches, encoded with different generator matrices: source packets belong to one or more generations, that use the same outer erasure code.

As in systematic RLNC, HSCNCs not only send encoded packets but also original data packets. The quantity of redundant packet can be optimised to improve the decoding error probability while not to significantly increase the overhead of the communication.

Figure 3.4 describes the general structure of the encoder of HSCNCs. Let’s consider a source, which has to send $k$ packets, each of which has $l$ symbols in $\mathbb{F}_q$ ($q = 2^m$): so each packet has $lm$ bits. The source information matrix is

$$ U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} $$

(3.3)

Next, the matrix $U$ is encoded via a HSCNC. At the first stage, a code $C(n, k)$ encodes the $k$ source packets into $n$ encoded ones: its output is an $n \times l$ matrix called $Y$. Then, at the second stage, these $n$ packets are encoded via systematic RLNC. In particular,
the packets encoded via RLNC are obtained by linearly combining the rows of matrix Y. Hence, the HSCNC encoder outputs

$$\begin{bmatrix} I & X \\ A \\ \end{bmatrix}$$

(3.4)

where I is the $n \times n$ identity matrix, X is the encoded matrix and A is the $\phi \times n$ matrix of random coefficients of the linear combinations. Matrix X is

$$X = \begin{bmatrix} Y \\ \hat{Y} \end{bmatrix}$$

(3.5)

where Y is the matrix encoded by the outer erasure code, and $\hat{Y}$ is the matrix of RLNC redundant encoded packets, generated via linear combinations.

The advantage of using systematic RLNC is that, even if RLNC is not successfully decoded, $C(n, k)$ can recover the lost information by decoding matrix Y or one of its received submatrices (if the erasures also affect the first $n$ rows). This cannot be possible with either RLNC or BATS codes: in fact, in both cases, a generation or a batch is decodable if and only if the sink can receive enough innovative linear combination of packets.

When RLNC is used, another aspect requires special attention: the overhead. In fact, RLNC needs packet tagging to send the random coefficients to the sink, in order to decode. The overhead introduced by packet tagging mainly depends on the size of the matrix $\hat{Y}$, which affects the size of matrix A. In particular, the overhead is $n$ symbols per redundant packet. The percentage of overhead per packet is obtained as $n/l$, which requires to be as small as possible.

The description and the evaluation of HSCNCs for different outer erasure codes follow. The evaluation will include the quantification of the decoding error probability, the minimisation of the coding rate, and the estimation of the encoding/decoding complexity.

In the following sections, we also provide theoretical analysis of decoding error probability of HSCNCs to justify the improvement these hybrid concatenated codes provide compared to pure RLNC. The simulation results in the remainder of this chapter have been obtained with MATLAB software. The simulation set up considers $\text{rank}(X) = 32$ and packet size $l = 64$ symbols.
3.2 Luby Transform Codes and Random Linear Network Coding

The first kind of HSCNC we study, is the one that has an LT code as outer code. The LT code gets the matrix $U$ of $k$ source row packets and transforms it into the matrix $U'$ as shown in figure 3.5. Then, LT encoder performs inter-packet encoding, so the output becomes a single-column matrix $Y'$ with $n \cdot l$ rows. This matrix is reshaped to obtain a matrix $Y$ of size $n \times l$. Finally, matrix $Y$ is encoded into matrix $X$ according to (3.5) by using systematic RLNC. This HSCNC has coding rate

$$R = R_{LT} \cdot R_{RLNC}$$ (3.6)

LT codes normally optimise performance and redundancy for sets of input symbols of size $k \cdot l > 10^6$. Since we handle quantities of input symbols that are $\leq 32l$ in the applications of this thesis, we have to carefully design the robust Soliton degree distribution to minimise the decoding error probability of the BP decoder. We evaluate this probability via MATLAB simulations. Nevertheless, the value of this probability can also be calculated either via numerical recursive simulations [54, 55] or can be determined by a rigorous expression only when the length of the code goes to infinity [55] (not applicable in our scenario for small $n$). However, a theoretical upper bound on the failure decoding probability of an LT code can be obtained by using the expression in [12] referred to a general random fountain code

$$P_{eLT} \leq 2^{-n+k}$$ (3.7)

which only depends on the size of the overhead $n-k$. In particular, the equality is roughly achieved when redundancy $n-k$ is not too small [12]. Let $R_{LT} = k/n_o$ be the coding rate of LT outer code, and let $R_{RLNC} = n_o/n_i$ be the coding rate of inner systematic RLNC. The analytical expression of decoding error probability $P_{eHSCNC}$ can become

$$P_{eHSCNC} = \begin{cases} P_{eLT}, & \text{if } \text{rank}(\tilde{X}) < n_o \\ P_{eRLNC} \cdot P_{eLT}, & \text{if } \text{rank}(\tilde{X}) > n_o \end{cases}$$ (3.8)
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.5: Structure of the input data matrix of LT encoder in a HSCNC. The matrix $U$ is transformed into matrix $U'$ by vertically concatenating the transpose of each packet.
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.6: Decoding error probability gain of HSCNCs with LT outer codes, taking as baseline RLNC. The overall coding rate is $R = 0.76$ and $R_{LT} = 0.77$, in order to verify the hypothesis in [12].

where $P_{e,LT}$ can be approximated to $2^{-n+k}$, if $n-k$ is not too small. However, this result can be considered an upper bound since it does not take into account the structure of the Robust Soliton distribution and the effect of burst erasures. Figure 3.6 shows an instance of this upper bound, when the coding rate of the HSCNC is $R \approx 0.76$. The result in represented in terms of gain, intended as the percentage of reduction of decoding error probability of the HSCNC compared to RLNC. This upper bound on the gain says that these HSCNCs can have lower decoding error probability than RLNC until the number of erasures is not very high.

In order to exactly analyse the decoding error probability of HSCNCs for burst erasure channel – taking into account the structure of the Robust Soliton distribution – we now move to MATLAB simulations. According to the characteristics of the robust Soliton distribution presented in subsection 2.3.3, we get the performance for three main trends. So, we select the values of parameters $\delta$ and $c$ and we test and compare the error decoding probability of HSCNCs with these different distributions. Moreover, the inner systematic RLNC encoder is tested by performing operations over $\mathbb{F}_4$ and $\mathbb{F}_{256}$:
in this way, we evaluate the impact of the decoding failure probability on the total error decoding probability.

First, figure 3.7 shows a robust Soliton distribution with \( \delta = 0.001 \) and \( c = 0.1 \). This distribution has two peaks: the first is lower than the second, hence the probability to have encoded symbols with higher degree is greater. After 100 trials to obtain the generator matrix, the calculated average degree is \( D_1 = 9.73 \). Second, figure 3.8 shows a robust Soliton distribution with \( \delta = 0.1 \) and \( c = 0.1 \). This distribution also has two peaks but the first is higher than the second: this means that there will be encoded symbols with lower degree with higher probability. After 100 trials to obtain the generator matrix, the calculated average degree is \( D_2 = 9.25 \). Third, figure 3.9 depicts a robust Soliton distribution with \( \delta = 4 \) and \( c = 0.1 \). This distribution has only one peak: hence the majority of encoded symbols will have small degree with high probability. After 100 trials to obtain the generator matrix, the obtained average degree is \( D_3 = 8.53 \). By comparing figure 3.7(b), figure 3.8(b) and figure 3.9(b), the differences among the outputs of these three distributions clearly appear. Since \( D_1 > D_2 > D_3 \), the expected number of operations per encoded symbol will be more in the first distribution than in the second and in the third. This will mainly affect encoding and decoding complexity. Moreover, figure 3.7(b) shows that the majority of symbols of this distribution have degree greater than 10, and that the probability of finding an encoded symbol with degree 1 is lower than the ones of distributions of figure 3.8(b) and figure 3.9(b). In particular, the third distribution is the one that has more symbols with lower degree and more probability to have encoded symbols with degree 1. After simulations with 100 trials for the generator matrix, the probability to find an encoded symbol with degree 1 in the three distributions is respectively 0.16, 0.22 and 0.25. As described in chapter 2, that significantly influences the behaviour and the performance of the BP decoder. In case of erasures, the third distribution has greater capability to keep presence of symbols with unitary degree than the first one.

After these premises, we simulate the performance of HSCNCs to compare their decoding error probability with LT outer codes following those three different distributions. The results will help us to identify the best characteristics LT codes need to be efficient outer codes for HSCNCs. In all the three configurations, packet size is set to \( l = 64 \)
Figure 3.7: Simulations to provide some insights into the characteristics of the Robust Soliton distribution (a) Robust Soliton distribution for $\delta = 0.001$ and $c = 0.1$. In order to keep the figure clear, the snapshot is limited to 50 on the horizontal axis (b) Distribution of the weight of each encoded symbol after 100 trials to obtain the generation matrix. The average degree is $D_1 = 9.73$. 
Figure 3.8: Simulations to provide some insights into the characteristics of the Robust Soliton distribution (a) Robust Soliton distribution for $\delta = 0.1$ and $c = 0.1$. In order to keep the figure clear, the snapshot is limited to 50 on the horizontal axis (b) Distribution of the weight of each encoded symbol after 100 trials to obtain the generation matrix. The average degree is $D_2 = 9.25$. 
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.9: Simulations to provide some insights into the characteristics of the Robust Soliton distribution (a) Robust Soliton distribution for $\delta = 4$ and $c = 0.1$. In order to keep the figure clear, the snapshot is limited to 50 on the horizontal axis (b) Distribution of the weight of each encoded symbol after 100 trials to obtain the generation matrix. The average degree is $D_3 = 8.53$. 
symbols and \( \text{rank}(X) = 32 \).

Figure 3.10 depicts the decoding error probability of HSCNCs in the first scenario, with LT codes with \( \delta = 0.001 \) and \( c = 0.1 \). Systematic RLNC performs operations over \( \mathbb{F}_4 \). The coding rate of LT codes is set to 0.8 and to 0.57. The coding rate of RLNC is set to 0.94 and to 0.91. This means that the coding rates of HSCNCs become: \( R_1 = 0.8 \cdot 0.94 = 0.75 \), \( R_2 = 0.57 \cdot 0.94 = 0.53 \), \( R_3 = 0.8 \cdot 0.91 = 0.73 \) and \( R_4 = 0.57 \cdot 0.91 = 0.52 \). Figure 3.10 clearly shows that this distribution performs very badly. In fact, even if the coding rate of the HSCNC is almost 1/2 the decoding error probability remains greater than 0.1. The decoding failure probability of BP decoder together with the one due to the small field size of systematic RLNC, causes a higher overall decoding error probability. Furthermore, the position of the degrees also is a negative aspect of this distribution (figure 3.7(b)): its symbols with high degree are neighbours; hence the erasure of a burst affects symbols from which several other symbols depend on. All these aspects together make the concatenation approach perform worse than RLNC alone.

Figure 3.11 depicts the decoding error probability of HSCNCs in the second scenario, with LT codes with \( \delta = 0.1 \) and \( c = 0.1 \). Systematic RLNC encoder performs operations over \( \mathbb{F}_4 \). The rates of LT codes and systematic RLNC are kept the same as before. This time, the codes with rate \( R_2 = 0.57 \cdot 0.94 = 0.53 \) and \( R_4 = 0.57 \cdot 0.91 = 0.52 \) performs better than before. The coding rate of HSCNC is greater than 1/2, and the decoding error probability remains zero until \( b_e < 0.15 \) for \( R = 0.53 \), and \( b_e < 0.2 \) for \( R = 0.52 \). The effect of the small field size has less impact on the decoding error probability because the outer code can recover more erasures.

This second HSCNC with LT code with \( \delta = 0.1 \) and \( c = 0.1 \) only outperforms the erasure-correcting capabilities of RLNC in \( \mathbb{F}_4 \), for coding rates of 0.53 and 0.52, so resulting in a significant reduction of coding rate compared to RLNC.

Finally, figure 3.12 shows the decoding error probability of HSCNCs in the third scenario with LT codes, with \( \delta = 4 \) and \( c = 0.1 \). Systematic RLNC performs operations over \( \mathbb{F}_4 \). The rates of LT codes and RLNC are kept the same as previously. These codes are the most effective: with rate \( R_2 = 0.57 \cdot 0.94 = 0.53 \) are successful for \( b_e < 0.2 \),
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.10: Decoding error probability of HSCNC constituted by an LT outer code with $\delta = 0.001$ and $c = 0.1$, and by inner systematic RLNC with $\mathbb{F}_4$.

Figure 3.11: Decoding error probability of HSCNC constituted by an LT outer code with $\delta = 0.1$ and $c = 0.1$, and by inner systematic RLNC with $\mathbb{F}_4$. 
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.12: Decoding error probability of HSCNC constituted by an LT outer code with $\delta = 4$ and $c = 0.1$, and by inner systematic RLNC with $\mathbb{F}_4$. The decoding error probability for $R_{LT} = 0.57$ and $R_{RLNC} = 0.91$ is always zero.

and for $b_e > 0.2$ always keeps the decoding error probability less than 0.01. The rate $R_4 = 0.57 \cdot 0.91 = 0.52$ is not depicted because its decoding error probability is always zero. Moreover, the average degree is lower than the one in the two previous scenarios. The effect of the small field size on the overall decoding error probability is significantly reduced: that is, this LT configurations also permits to keep low the complexity of RLNC encoding/decoding operations. By comparing figure 3.8(b) and figure 3.9(b), it is possible to see that the density of symbols with high degree in the second distribution is much reduced than the one in the first, and in particular there are very few neighbours with high degree. Hence, we can say that this last distribution provides efficient and effective LT outer codes for HSCNCs with RLNC over $\mathbb{F}_4$, to correct burst erasures. By generalising this result, a distribution needs to have low average degree and to avoid neighbours with high degree.

Now, it is useful to compare the results obtained above for inner systematic RLNC over $\mathbb{F}_4$, to the ones calculated when the systematic RLNC performs operations over
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.13: Decoding error probability of HSCNC constituted by an LT outer code with $\delta = 0.001$ and $c = 0.1$, and by inner systematic RLNC with $F_{256}$. This allows us to quantify the impact of the failure probability of systematic RLNC on this configuration of HSCNCs. According to expression 2.15, the decoding failure probability for $F_4$ and for $F_{256}$ are upper bounded respectively by $P_e \leq 0.03$ and $P_e \leq 6.03 \cdot 10^{-8}$. Then, when the finite field is $F_{256}$, $P_e$ becomes negligible.

Figure 3.13 compares HSCNCs with systematic RLNC in $F_4$ and $F_{256}$, with LT codes following the first distribution. Even if the decoding error probability remains poor (always $> 0.01$), the image shows that the degradation due to the decoding failure probability of RLNC in the first scenario is between 0.1 and 0.2.

However, the impact of the decoding failure probability is more effective when LT codes follow the second and the third scenario. Figure 3.14 reveals that RLNC operations over $F_{256}$ shift the successful decoding from $b_e = 0.15$ to $b_e = 0.23$ by keeping the same coding rate $R_2 = 0.53$, and from $b_e = 0.2$ to $b_e = 0.23$ if the coding rate is $R_4 = 0.52$. Moreover, at the same $b_e$, the curves in $F_4$ display a high degradation: in fact, the decoding error probability in $F_4$ is almost 10 times higher than the one in $F_{256}$ for
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.14: Decoding error probability of HSCNC constituted by an LT outer code with $\delta = 0.1$ and $c = 0.1$, and by inner systematic RLNC with $F_{256}$.

$R_2 = 0.53$, and 5 times higher for $R_4 = 0.52$.

Nevertheless, the third configuration of the robust Soliton distribution appears to be the less affected by the decoding failure probability of RLNC. If the LT outer code has rate $R_{LT} = 0.71$, the decoding error probability is zero for $b_e < 0.15$. The curve in $F_4$ with LT code with rate $R_{LT} = 0.57$ was successful for $b_e < 0.2$. This demonstrates that the change in the size of the finite field of RLNC permits a reduction of redundancy of 0.14. Moreover, the curves for LT codes with $R_{LT} = 0.57$ and RLNC in $F_{256}$ are not depicted since their decoding is always successful in the range $0.01 \leq b_e \leq 0.3$.

Finally, let’s analyse the complexity and the overhead of the serial concatenation of LT codes and systematic RLNC. As discussed in subsection 2.3.3, the encoding and decoding complexity of LT codes are the same, and linearly depend on the average degree $D$ and on the number of symbols in matrix $Y$, that is $nl$. So, the main parameters that influence the complexity of the encoding and decoding operations of these HSCNCs are: the average degree of LT codes $D$, the size of the information generated by LT
Figure 3.15: Decoding error probability of HSCNC constituted by an LT outer code with $\delta = 4$ and $c = 0.1$, and by inner systematic RLNC with $\mathbb{F}_{256}$. 

encoder, and the size of the redundancy of RLNC. The encoding and decoding delay of these HSCNCs has been simulated by using the MATLAB function `tic/toc`. Figure 3.16 represents the average values calculated after 50 iterations. The output matrix is a matrix of $n = 32$ with $l = 64$ symbols in $\mathbb{F}_{256}$. The hardware used in the simulations was a processor Intel Core i3-2310M with frequency 2.10 GHz. The results can give a qualitative idea in the difference of complexity among the HSCNCs with the three LT outer codes, by considering systematic RLNC in $\mathbb{F}_{4}$ and $\mathbb{F}_{256}$. These simulations confirm the efficiency of the LT outer codes with distribution $\delta = 4$ and $c = 0.1$, because its average degree is the lowest and because the frequency of symbols with degree 1 is the highest.
3.2. Luby Transform Codes and Random Linear Network Coding

Figure 3.16: HSCNC encoding and decoding time performance with LT outer codes. The results are the average of 50 iterations. $R$ considers the rates of LT codes and $GF$ is the finite field of systematic RLNC (a) Encoding time of HSCNCs with LT outer codes with $\delta = 0.001$, $\delta = 0.1$ and $\delta = 4$ (b) Decoding time of HSCNCs with LT outer codes with $\delta = 0.001$, $\delta = 0.1$ and $\delta = 4$. 
3.3 Reed-Solomon Codes and Random Linear Network Coding

Another family of HSCNCs is the one constituted by RS outer codes and inner systematic RLNC. In particular, we use FRS outer codes in the work of this thesis, with Berlekamp-Massey decoder.

Given as input matrix $U$ of $k$ source packets, with $l$ symbols, the outer code becomes an FRS$_q^{(l)}[n, k]$ code, with $q = 2^m$, block length $n/l$ and rate $R_{FRS} = k/n$. Since $m = 8$ is the value generally used in RS applications, all the following discussion will consider FRS$_{256}^{(l)}[n, k]$ codes in $\mathbb{F}_{256}$. The coding rate of this HSCNC is

$$R = R_{RS} \cdot R_{RLNC}$$

(3.9)

Let $R_{RS} = k/n_o$ be the coding rate of RS outer code, and let $R_{RLNC} = n_o/n_i$ be the coding rate of inner systematic RLNC. The exact decoding error probability of a RS code can be calculated [56] as

$$P_{e_{RS}} = \left( \frac{n_o}{d_{min}} \right) b_e^{d_{min}} (1 - b_e)^{(n_o - d_{min})} + \left( \frac{n_o}{d_{min} + 1} \right) b_e^{(d_{min} + 1)} (1 - b_e)^{(n_o - d_{min} - 1)} + \left( \frac{n_o}{d_{min} + 2} \right) b_e^{(d_{min} + 2)} (1 - b_e)^{(n_o - d_{min} - 2)} + \ldots$$

(3.10)

where $d_{min} = n_o - k + 1$. So, the decoding error probability $P_{e_{HSCNC}}$ for a HSCNC that uses RS outer codes is

$$P_{e_{HSCNC}} = \begin{cases} P_{e_{RS}}(b'_e), & \text{if } \text{rank}(\hat{X}) < n_o \\ P_{e_{RLNC}} \cdot P_{e_{RS}}, & \text{if } \text{rank}(\hat{X}) > n_o \end{cases}$$

(3.11)

where $b'_e$ is the burst erasure probability that erases packets related to RS code. In order to have an upper bound on the possible gain in terms of reduction of decoding error probability, we set

$$b'_e = \frac{[b_e n_i] - n_i + n_o}{n_o}$$

(3.12)

In particular, the gain of the HSCNC is defined as in the previous section. The coding rate of baseline RLNC and of the HSCNC is chosen approximately equal, so that
we can see the improvement in terms of error correction given the same redundancy. Figure 3.17 depicts the decoding error probability gain of HSCNCs with RS outer codes, taking as baseline RLNC. The two codes have same coding rate $R \approx 0.76$ and $R \approx 0.85$. Theoretically, it is possible to see that this family of HSCNCs provides a significant higher error-correcting capability given the same redundancy. Especially, by decreasing the coding rate for both codes the gain increases. Furthermore, by augmenting $b_e$ at fixed coding rates, the gain decreases since the decoding error probability goes to 1 for both codes.

After these premises via theoretical analysis, let’s start the study of the decoding error probability of this kind of HSCNCs via MATLAB simulations. In all the simulations below, the size of data packets remains $l = 64$, and $\text{rank}(X) = 32$. As in the previous section, we first start the description of the HSCNCs with systematic RLNC over $\mathbb{F}_4$. Figure 3.18 depicts the decoding error probability of HSCNC with inner systematic RLNC with $R_{RLNC} = 0.94$. The coding rate of FRS is kept very high, since they are
MDS codes and are very effective in case of burst erasures: especially, the results take into account FRS codes with coding rate $R_{RS} = 0.94$, $R_{RS} = 0.9$ and $R_{RS} = 0.79$. Hence, the coding rates of the simulated HSCNCs are respectively $R_1 = 0.94^2 = 0.88$, $R_2 = 0.9 \cdot 0.94 = 0.84$, and $R_3 = 0.79 \cdot 0.94 = 0.74$. It is possible to notice that for $b_e < 0.05$ the decoding error probability is less than 0.01 or null. This means that the decoding failure probability of RLNC has significant impact on the decoding error capability of the outer decoder. Even so, the FRS outer decoder performs better than the LT one (see figure 3.10).

Figure 3.19 shows the decoding error probability when inner systematic RLNC has lower coding rate, $R = 0.91$. By maintaining the same rates for FRS codes, the HSCNCs have rates $R_1 = 0.94 \cdot 0.91 = 0.85$, $R_2 = 0.9 \cdot 0.91 = 0.82$, and $R_3 = 0.79 \cdot 0.91 = 0.72$. With these configurations, HSCNCs with rates $R_1$ and $R_2$ guarantee a decoding error probability $\leq 0.01$ for $b_e < 0.1$, while HSCNCs with rate $R_3$ achieve a decoding error probability $\leq 0.01$ for $b_e < 0.15$: this means that a reduction of 0.03 in coding rate of RLNC permits efficient erasure recovery for 0.05 more in burst erasure probability. An increase in the redundancy $\phi$ of RLNC decreases the decoding failure probability $P_e$: the comparison between figure 3.18 and figure 3.19 underlines that in fields of small size, the redundancy of RLNC needs to be augmented to achieve reliability.

After having analysed inner RLNC over $\mathbb{F}_4$, we go on in the study of HSCNCs when systematic RLNC performs operations over $\mathbb{F}_{256}$. In this case, the effect of the decoding failure probability becomes almost negligible. In fact, the significant improvement appears clear by comparing figure 3.20 to figure 3.18. Figure 3.20 draws the decoding error probability of HSCNCs with inner systematic RLNC with $R_{RLNC} = 0.94$. All the three rates for different FRS configurations guarantee a successful decoding for $b_e < 0.15$. Since the difference in decoding error probability among the different coding rates of FRS outer codes are not effectively improving the results, a HSCNC with rate 0.88 is enough to have acceptable burst-erasure correction for $b_e < 0.15$.

Figure 3.21 shows the decoding error probability of HSCNCs with inner systematic RLNC with $R_{RLNC} = 0.91$. As in figure 3.20, the increase in redundancy for the FRS codes does not significantly augment the decoding error probability of the HSCNC.
3.3. Reed-Solomon Codes and Random Linear Network Coding

Figure 3.18: Decoding error probability of HSCNC constituted by an RS outer code and by inner systematic RLNC over $\mathbb{F}_4$, with $R_{RLNC} = 0.94$.

Figure 3.19: Decoding error probability of HSCNC constituted by an RS outer code and by inner systematic RLNC over $\mathbb{F}_4$, with $R_{RLNC} = 0.91$. 
3.3. Reed-Solomon Codes and Random Linear Network Coding

Figure 3.20: Decoding error probability of HSCNC constituted by an RS outer code and by inner systematic RLNC over $\mathbb{F}_{256}$, with $R_{RLNC} = 0.94$.

Moreover, the improvement due to RLNC operations over $\mathbb{F}_{256}$ becomes clear if figure 3.19 is compared to figure 3.21. This latter configuration of HSCNCs with rate $R = 0.85$ guarantees a successful decoding for $b_e \leq 0.26$. The examination of these codes reveals that they can outperform all the previous schemes (RLNC and HSCNCs with LT outer codes) presented in this chapter.

Finally, we discuss the complexity of these HSCNCs. The encoding complexity of FRS outer codes depends on the redundancy and on the size of output data as $O(n(n-k))$ [57]. On the other hand, their decoding complexity has quadratic dependence on redundancy as $O(2(n-k)^2)$ [57] (Berlekamp-Massey algorithm). Figure 3.22 represents the results of simulations of encoding (figure 3.22(a)) and decoding (figure 3.22(b)) time of HSCNCs with FRS outer codes. The parameters considered in the simulations are packet size $l = 64$ symbols, and encoded matrix $Y$ with $n = 32$ rows.

In general, the analysis and the simulations above demonstrate the importance for HSCNCs to have MDS codes as outer codes, in order to achieve the highest performance.
3.4. Low Density Parity Check Codes and Random Linear Network Coding

This section explains some results to show the advantages and the drawbacks of HSCNCs constituted by LDPC outer codes and systematic RLNC. The LDPC encoder encodes the source information packets of matrix $U$ by reshaping them into a stream, as described in figure 3.5 for LT encoder. Then, after encoding and the reshaping of $Y'$, matrix $Y$ is given as input to the inner systematic RLNC encoder. The coding rate of this concatenated coding scheme is

$$R = R_{LDPC} \cdot R_{RLNC}$$  \hspace{1cm} (3.13)

The exact analytical expression of the decoding error probability of LDPC codes using message passing decoder in case of burst erasure channels is not possible because this decoding technique has a nonlinear behaviour [58]. As for LT codes with BP decoder, there are only approximate recursive solutions based on finite-length analysis [59].

The results in [8] claim that the parity-check matrix with structure (2.4) following
Figure 3.22: HSCNC encoding and decoding time performance with FRS outer codes. The results are the average of 50 iterations. $R$ considers the rates of FRS codes and $GF$ is the finite field of systematic RLNC (a) Encoding time of HSCNCs with FRS outer codes with $R = 0.94$, $R = 0.91$ and $R = 0.79$ (b) Decoding time of HSCNCs with FRS outer codes with $R = 0.94$, $R = 0.91$ and $R = 0.79$. 
3.4. Low Density Parity Check Codes and Random Linear Network Coding

Figure 3.23: Decoding error probability of HSCNC constituted by an LDPC outer code and by inner systematic RLNC over $\mathbb{F}_4$, with $R_{RLNC} = 0.94$ and $R_{RLNC} = 0.91$. Then, the final rates of the two HSCNCs respectively are $R_1 = 0.5 \cdot 0.94 = 0.47$ and $R_2 = 0.5 \cdot 0.91 = 0.45$. In particular, the code with rate $R_2$ is the most efficient since it can guarantee reliability for $b_e \leq 0.14$. The negative effect of the decoding failure probability of RLNC in $\mathbb{F}_4$ heavily affects the performance of the
3.4. Low Density Parity Check Codes and Random Linear Network Coding

Figure 3.24: Decoding error probability of HSCNC constituted by an LDPC outer code and by inner systematic RLNC over $\mathbb{F}_{256}$, with $R_{RLNC} = 0.94$ and $R_{RLNC} = 0.91$. So, the final rates remain the same of the ones in figure 3.23. This time, the parameter that changes is only the size of the finite field of inner systematic RLNC, which performs operations over $\mathbb{F}_{256}$. The decoding failure probability $P_e$ becomes negligible and the results clearly shows the significant improvement in erasure correction. Especially, the simulations demonstrate the weakness of these LDPC codes for higher values of $P_e$, hence the decoding failure probability of RLNC results to be determinant for the overall performance of the concatenated code. The HSCNCs are both always reliable in the range $0.01 \leq b_e \leq 0.4$. Furthermore, the HSCNC with rate $R_1$ is successful for $b_e < 0.2$, while the one with rate $R_2$ is successful for $b_e < 0.36$.

Finally, we analyse the complexity of this concatenated coding scheme. Figure 3.25 displays the average results after 50 runs. The encoding and decoding time of the
3.5 Raptor Codes and Random Linear Network Coding

Figure 3.25: HSCNC encoding and decoding time performance with LDPC outer codes with $R_{LDPC} = 0.5$. The results are the average of 50 iterations. $GF$ is the finite field of systematic RLNC.

LDPC outer code are almost the same, and depend on the weight of the columns of the parity-check matrix $H$; hence on the number of nonzero elements in polynomials (2.7). As previously explained, systematic RLNC complexity is due to $\phi$ and $q$. In the special case of our simulations $\phi$ only varies for rates $R_{RLNC} = 0.94$ and $R_{RLNC} = 0.91$, then what mainly influences the encoding and decoding time is the size of the finite field. In fact, when $q_{RLNC} = 256$ the encoding time is 41 percent more than when $q_{RLNC} = 4$. On the other hand, when $q_{RLNC} = 256$ the decoding time is 53 percent more than when $q_{RLNC} = 4$.

3.5 Raptor Codes and Random Linear Network Coding

The last configuration we study in this chapter is HSCNCs constituted by Raptor outer codes and inner systematic RLNC. In particular, we analyse two configurations for Raptor codes, one using LDPC pre-coding, and another one using FRS pre-coding. While the former is the one suggested by the literature, to the best of our knowledge, the latter has never been proposed and tested. In particular, we evaluate its performance via MATLAB simulations. RS code are always seen as antagonist of Raptor codes...
because of their greater complexity and less flexibility [61]. However, it is interesting in our context to evaluate how the concatenation of RS codes and LT codes can improve systematic RLNC in HSCNCs.

3.5.1 Raptor Codes with Low Density Parity Check Codes

The HSCNC that has Raptor code as outer code, is obtained by the serial concatenation of three erasure codes: LDPC code, LT code and systematic RLNC. So, the rate can be expressed as

$$R = R_{LDPC} \cdot R_{LT} \cdot R_{RLNC}$$

The simulations of this scheme consider packet size $l = 64$ symbols, $\text{rank}(X) = 32$ and $R_{LT} = 0.91$. The value of $R_{LT} = 0.91$ has been chosen to minimise the coding rate while keeping low the decoding failure probability of LT decoder. Moreover, because of its effectiveness and efficiency, the configuration with $\delta = 4$ and $c = 0.1$ has been selected. The LDPC code has rate 0.5 and RLNC has rates 0.94 and 0.91. Then, the rates of HSCNCs are respectively 0.42 and 0.41. After simulations, the schemes can always achieve successful decoding for the range $0.01 \leq b_e \leq 0.3$.

Figure 3.26 shows the simulation results of the encoding and decoding time of HSCNCs with Raptor outer codes and with LDPC pre-coding. The average values calculated after 50 iterations compared to ones in figure 3.25, demonstrate that the delay of encoding/decoding operations is not significantly increased by LT encoding/decoding time.

3.5.2 Raptor Codes with Reed-Solomon Codes

The last coding scheme we examine in this chapter is HSCNC constituted by a novel somewhat Raptor outer code that has FRS pre-coding. The overall rate of the concatenated code is defined as

$$R = R_{RS} \cdot R_{LT} \cdot R_{RLNC}$$

As we have seen in section 3.2, there is not exact analytical expression for decoding error probability for LT codes. So, it is only possible to have an upper bound on the
3.5. Raptor Codes and Random Linear Network Coding

Figure 3.26: HSCNC encoding and decoding time performance with Raptor outer codes with $R = 0.46$. The results are the average of 50 iterations. $GF$ is the finite field of systematic RLNC.

performance by using the general expression for a random fountain (see (3.7)). Let $R_{RS} = k/n_1$ be the coding rate of RS pre-code, let $R_{LT} = n_1/n_o$ be the coding rate of the LT outer code, and let $R_{RLNC} = n_o/n_i$ be the coding rate of inner systematic RLNC. Thus, the decoding error probability $P_{ehscnc}$ becomes

$$P_{ehscnc} = \begin{cases} P_{eRS}(b_e') \cdot P_{eLT}, & \text{if rank}(\bar{X}) < n_o \\ P_{eRLNC} \cdot P_{eLT} \cdot P_{ERS}, & \text{if rank}(\bar{X}) > n_o \end{cases}$$

(3.16)

where $b_e'$ is defined in equation (3.12). Figure 3.27 depicts the gain of this family of HSCNCs in respect of RLNC, as defined in section 3.2. The coding rates are approximately $R \approx 0.85$ and $R \approx 0.76$. We can see that this hybrid concatenated scheme can outperform RLNC, and is particularly effective (compared to HSCNCs with both LT and RS outer codes) when burst erasure probability is high.

Since this sort of Raptor codes have never been studied in the literature, we check the performance according to all three LT configurations, identified in section 3.2.

Figure 3.28 depicts the decoding error probability of HSCNCs constituted by Raptor outer codes ($R_{LT} = 0.91$, $\delta = 0.001$ and and RS pre-coding) and by inner systematic
Figure 3.27: Decoding error probability gain of HSCNCs with Raptor outer codes (RS pre-coding), taking as baseline RLNC. The coding rate of RLNC and the overall coding rate of HSCNCs are $R \approx 0.76$ and $R \approx 0.85$. 
3.5. Raptor Codes and Random Linear Network Coding

Figure 3.28: Decoding error probability of HSCNC constituted by Raptor outer codes \( (R_{LT} = 0.91, \delta = 0.001 \text{ and RS pre-coding}) \) and by inner systematic RLNC over \( \mathbb{F}_4 \), with \( R_{RLNC} = 0.94 \) and \( R_{RLNC} = 0.91 \).

RLNC over \( \mathbb{F}_4 \), with \( R_{RLNC} = 0.94 \) and \( R_{RLNC} = 0.91 \). The rates of the simulated HSCNCs become \( R_1 = 0.9 \cdot 0.91 \cdot 0.94 = 0.77 \), \( R_2 = 0.86 \cdot 0.91 \cdot 0.94 = 0.73 \), \( R_3 = 0.9 \cdot 0.91^2 = 0.74 \) and \( R_4 = 0.86 \cdot 0.91^2 = 0.71 \). The two codes with rate \( R_3 \) and \( R_4 \) are respectively successful for \( 0.01 \leq b_e \leq 0.15 \) and \( 0.01 \leq b_e \leq 0.13 \). By comparing this scenario with the one in figure 3.10, the improvement appears clear. The HSCNC in figure 3.10 was not reliable even when the coding rate was achieving \( R = 0.52 \). On the other hand, with this configuration for Raptor outer codes that use the same distribution for the LT code, rates such as \( R_3 = 0.74 \) and \( R_3 = 0.71 \) are enough to achieve reliability for \( b_e \leq 0.15 \).

Figure 3.29 shows the decoding error probability of HSCNC constituted by Raptor outer codes \( (R_{LT} = 0.91, \delta = 0.1 \text{ and RS pre-coding}) \) and by inner systematic RLNC over \( \mathbb{F}_4 \), with \( R_{RLNC} = 0.94 \) and \( R_{RLNC} = 0.91 \). The HSCNCs keep the same coding rates. This time it is curious to see a behaviour that is different from what was happening in figure 3.11. In fact, the distribution that before was improving the reliability of the code, now appears to be less reliable.
3.5. Raptor Codes and Random Linear Network Coding

Figure 3.29: Decoding error probability of HSCNC constituted by Raptor outer codes \((R_{LT} = 0.91, \delta = 0.1\) and RS pre-coding) and by inner systematic RLNC over \(\mathbb{F}_4\), with \(R_{RLNC} = 0.94\) and \(R_{RLNC} = 0.91\).

The same thing happens in figure 3.30. The performance of HSCNCs with the LT distribution \(\delta = 4\) and \(c = 0.1\), that in figure 3.12 was the most reliable, now results to be less effective for erasure correction than the one with \(\delta = 0.001\).

After the analysis of HSCNCs with inner systematic RLNC performing operations over \(\mathbb{F}_4\), we study the case, in which the finite field of RLNC is \(\mathbb{F}_2^{56}\) and the decoding failure probability of RLNC becomes negligible. The coding rates of HSCNCs remain the same as before. Figure 3.31 depicts the decoding error probability of HSCNCs constituted by Raptor outer codes \((R_{LT} = 0.91, \delta = 0.001\) and and RS pre-coding), and by inner systematic RLNC with \(R_{RLNC} = 0.94\) and \(R_{RLNC} = 0.91\). The concatenated codes with rates \(R_1\) and \(R_2\) are successful for \(b_e < 0.2\), while the ones with rates \(R_3\) and \(R_4\) are successfully correcting erasures in the range \(0.01 \leq b_e \leq 0.3\).

Figure 3.32 and figure 3.33 respectively displays the decoding error probability of HSCNC, constituted by Raptor outer codes \((R_{LT} = 0.91\) and RS pre-coding) and by inner systematic RLNC, with \(R_{RLNC} = 0.94\) and \(R_{RLNC} = 0.91\), when \(\delta = 0.1\).
3.5. Raptor Codes and Random Linear Network Coding

Figure 3.30: Decoding error probability of HSCNC constituted by Raptor outer codes \((R_{LT} = 0.91, \delta = 4\) and RS pre-coding) and by inner systematic RLNC over \(F_4\), with \(R_{RLNC} = 0.94\) and \(R_{RLNC} = 0.91\).

and \(\delta = 4\). Both configurations achieve successful decoding for \(b_e < 0.2\). Moreover, the HSCNCs with coding rates \(R_3\) and \(R_4\) are always successful in the rage \(0.01 \leq b_e \leq 0.3\).

Finally, figure 3.34 shows the encoding (figure 3.34(a)) and decoding (figure 3.34(b)) time of HSCNCs with with Raptor (FRS pre-coding) outer codes of rates \(R_{RS} = 0.9\) and \(R_{RS} = 0.86\), for systematic RLNC performing operations over \(F_4\) and \(F_{256}\). The results are also provided for the three robust Soliton distributions \(\delta = 0.001\), \(\delta = 0.1\) and \(\delta = 4\). Since the complexity of LT codes is much less than that of RS codes and RLNC, the encoding/decoding time of these codes is comparable to the one presented in figure 3.22, for similar coding rates.
3.5. Raptor Codes and Random Linear Network Coding

Figure 3.31: Decoding error probability of HSCNCs constituted by Raptor outer codes ($R_{LT} = 0.91$, $\delta = 0.001$ and RS pre-coding) and by inner systematic RLNC over $\mathbb{F}_{256}$, with $R = 0.94$ and $R = 0.91$.

Figure 3.32: Decoding error probability of HSCNC constituted by Raptor outer codes ($R_{LT} = 0.91$, $\delta = 0.1$ and RS pre-coding) and by inner systematic RLNC over $\mathbb{F}_{256}$, with $R_{RLNC} = 0.94$ and $R_{RLNC} = 0.91$. 
3.6 Summary and Conclusion

This chapter has provided a novel solution to improve the burst erasure-correcting capabilities of RLNC. We have designed serial concatenated coding schemes called HSCNCs, which are based on a serial concatenation of a ‘classical’ erasure outer code and inner systematic RLNC. These hybrid concatenated codes are the enabler to find effective and efficient coding solutions for the proposed new protocol EC-MIH (see chapter 4) in order to provide reliable VHOs by using RLNC.

We have tested the performance of HSCNCs with the following outer codes: LT codes, RS codes, LDPC codes and Raptor codes. In particular, two schemes for Raptor codes have been implemented: the first with LDPC pre-coding, already proposed in the literature, and the second with RS pre-coding, that is presented for the first time here. The analysis has been done via MATLAB simulations, supported by theoretical work. Given the overall results in this chapter, we can conclude that there are certain HSCNCs that can outperform RLNC in terms of decoding error probability.

Figure 3.33: Decoding error probability of HSCNC constituted by Raptor outer codes ($R_{LT} = 0.91$, $\delta = 4$ and RS pre-coding) and by inner systematic RLNC over $\mathbb{F}_{256}$, with $R_{RLNC} = 0.94$ and $R_{RLNC} = 0.91$. 

3.6 Summary and Conclusion
3.6. Summary and Conclusion

Figure 3.34: HSCNC encoding and decoding time performance with Raptor (FRS pre-coding) outer codes. The results are the average of 50 iterations. $R$ considers the rates of FRS codes and $GF$ is the finite field of systematic RLNC (a) Encoding time of HSCNCs with FRS pre-coding with $R = 0.9$ and $R = 0.86$ (b) Decoding time of HSCNCs with FRS pre-coding with $R = 0.9$ and $R = 0.86$. 
3.6. Summary and Conclusion

The study of expression (3.1) showed the degradation of decoding error probability of RLNC by increasing the burst erasure probability $b_e$. Moreover, the impact on the decoding error probability of the failure decoding probability due to the size of the finite field, has been underlined. In the results, we can distinguish the range of $b_e$ in which the code is reliable, and the range of $b_e$ in which the code is completely successful.

Since the focus of this thesis is to provide efficient VHOs, we go on with the analysis by considering the constraint of reliability.

Let’s now start the comparison with HSCNCs with LT outer codes. The robust Soliton distribution that is used to generate the generator matrix of the code should have either $\delta > 1$ or $\delta \leq 10^{-9}$, and $c = 0.1$. The other possibility is $\delta = 0.1$ and $c \geq 0.3$. The particular case we have simulated is $\delta = 4$ and $c = 0.1$. The HSCNCs with this configuration are always reliable in $0.01 \leq b_e \leq 0.3$ for $R \leq 0.53$, when systematic RLNC is over $\mathbb{F}_4$. Next, when systematic RLNC is over $\mathbb{F}_{256}$, the HSCNC is reliable in $b_e < 0.15$ for $R = 0.67$, and always reliable for $R \leq 0.53$.

The test of HSCNCs with FRS outer codes, only show significant performance for systematic RLNC over $\mathbb{F}_{256}$. The coding rate $R = 0.88$ guarantees reliability for $b_e < 0.15$, while the coding rate $R = 0.85$ provides reliability for $b_e \leq 0.26$.

When HSCNCs use LDPC outer code, they can achieve reliability in $0.01 \leq b_e \leq 0.3$ with systematic RLNC over $\mathbb{F}_{256}$. Nevertheless, the drawback of the scheme in reference [8] is that the rate of HSCNC is $R = 0.47$. For VHO applications, in which latency is critical, the coding rate should be greater than 0.5.

Due to the previous considerations, we cannot suggest the use of Raptor codes with LDPC pre-coding presented in this chapter. They always result to be reliable but introduce too much redundancy ($R < 0.5$). On the other hand, we did find effective HSCNCs that exploit as outer code our newly proposed 'somewhat' Raptor code in synergy with FRS pre-coding. In this case, the robust Soliton distribution that achieves the best performance is the one with $\delta = 0.001$ and $c = 0.1$. When systematic RLNC performs operations over $\mathbb{F}_4$, the HSCNC is reliable for $b_e < 0.16$ with coding rate $R = 0.82$. Instead, when systematic RLNC performs operations over $\mathbb{F}_{256}$, the HSCNC is reliable for $b_e < 0.2$ with coding rate $R = 0.85$, and in all the range of $0.01 \leq b_e \leq 0.3$.
if the coding rate is $R \leq 0.82$.

By analysing the complexity and the latency that encoding and decoding requires, the most efficient HSCNCs are the ones that use LT codes and FRS codes as outer codes. LDPC codes are also very efficient, but the coding rate they require is not acceptable for our applications.

Finally, we conclude by citing the schemes which we consider effective and efficient for EC-MIH applications, according to both theoretical and simulation results. We can claim that HSCNCs which exploit FRS outer codes significantly outperform RLNC over $\mathbb{F}_{256}$. At approximately equal coding rates, these HSCNCs provide a significant gain in terms of burst erasure probability. Together with these, the HSCNCs with the novel Raptor outer codes are more effective than RLNC at approximately equal coding rates. Especially, their performance is better than other HSCNCs for higher burst erasure probability. The third scheme we may suggest is the one that use LT outer codes. Even if they require lower coding rates to be more effective, LT codes are the most efficient in terms of latency and complexity.
Chapter 4

Hybrid Coding for Enhanced-Coded MIH (EC-MIH) Protocol

Chapter Outline: this chapter describes and evaluates the performance of the novel designed EC-MIH protocol. The first aim of the chapter is to justify the application of erasure-correcting codes to improve performance during VHOs. In order to do that, we show how the losses during VHOs behave according to the transmission rate and to the number of MNs in the network. In particular, we provide accurate functions to characterise the losses of MIH scenarios. Next, we develop the general architecture of EC-MIH protocol, new primitives, and present the changed logical scheme and the modified packet structure. The chapter ends with an analysis of the protocol overhead due to the built-in coding operations. Finally, we evaluate the performance of EC-MIH protocol in terms of the achieved average throughput and the VHO time, and compare this against the baseline MIH approach.

4.1 Introduction

Chapter 2 described the principal characteristics of IEEE 802.21 standard. Its main objective is to allow mobile users to perform seamless VHOs, and to improve the
user experience of mobile terminals. When MIH architecture and MIH protocol are deployed, packets are lost especially because of address resolution issues, handover operations and insufficient bandwidth in the radio access technologies. These losses significantly affect the quality of mobile user communications, for either audio or video. In particular, for audio, erased data cause choppy and broken audio. So, reference [62] recommends to keep erasure probability less than 2% for mobile voice. On the other hand, for video in mobile scenarios, the recommended erasure probability is less than 1% for non-interactive video, and less than 0.1% for interactive video.

These premises reveal that packet loss is a central aspect to handle in order to provide seamless VHOs. IEEE 802.21 standard guarantees the required reliability by exploiting an acknowledgement service together with retransmissions. As we discussed in the previous chapters, an MIH transmission is reliable if the loss probability is less than 1%. Nevertheless, the erasures can involve acknowledgements and retransmissions as well, thus the quality of user experience and the efficiency of the communication can be heavily affected.

An alternative solution to ARQ, to handle errors due to erasures, is the use of FEC. In fact, if the channel conditions become increasingly worse, an ARQ system maintains reliability while becoming less efficient. This motivates a new MIH protocol design known as EC-MIH to manage burst erasures during VHOs more effectively. In particular, the possibility of exploiting erasure-correcting codes comes from the capability of modelling VHOs as burst erasure channels. The correctness of seeing VHOs as burst erasure channels comes from the analyses and the simulations in [27, 63].

Another novel aspect of EC-MIH protocol is the exploitation of HSCNCs to enhance the quality of audio and/or video communication during VHOs. However, the novel protocol has an inherent structure which provides flexibility to include and deploy any kind of erasure code. When FEC is applied, the augmentation of losses, at fixed coding rate, makes the communication less and less reliable. So, our results and conclusions from chapter 3 are important to select the correct HSCNC to make the new EC-MIH protocol efficient and effective.
4.1. Introduction

Figure 4.1: Example of MIHF communication model.

4.1.1 A Model for Vertical Handovers

In order to estimate and handle erasures effectively through FEC, it is important to understand how the errors occur. Therefore, the design of an accurate channel model is pivotal in order to find specific codes to correct particular error distributions. Since it is hard to analytically define channel with memory in terms of characteristics of the individual causes of errors, it is more convenient to use sampled sequences, obtained either via simulations or measurements. Moreover, the quantity of samples indicates the degree of complexity of the channel model and accuracy. In our following analysis, we try to find an accurate structure of the channel, while keeping the complexity low.

First of all, we can claim that a Gilbert model is a possible accurate analytical description for losses during VHOs. The studies and the results presented in [27, 63], confirm that packet loss during VHOs can be modelled as a channel with memory (burst erasure channel).

Let us first consider the study of VHOs in [63]. The network topology of the MIHF communication model is represented in figure 4.1. IEEE 802.21 standard support both hard and soft VHOs depending on whether the handover procedure is break-before-make or make-before-break. In case of hard VHOs, if we follow the model in [63], the MIHF scenario depicted in figure 4.1 becomes the one in figure 4.2. Since there is only one link active at a time between the MN and the PoAs, we can translate the scenario as a line network with N intermediate nodes between source and sink. In this scenario,
4.1. Introduction

Figure 4.2: Example of MIHF communication model for hard VHOs. As it is possible to see, the scenario can be modelled as a line network.

Figure 4.3: Example of MIHF communication model for soft VHOs. If there is only an intermediate node, the scenario becomes a diamond network.

the losses are due to the time required by the switch to pass from 'ON' to OFF, and to 'ON' in the other line network. Furthermore, figure 4.3 describes how the model for soft VHOs becomes according to [63]. In this scenario, both the connections are active. Anyway, there is a time interval, in which the quality of the connection is so bad that there is a percentage of packets lost. The duration of this interval mainly influences the packet loss probability.

In such scenarios, the burst erasure probability is represented by a two-variable function. During a hard VHO, all the packets transmitted during the lack of connection are lost. As we said, the link between the MIH source and the first node of the line network behaves like a switch. The time \( t \) for the switch to move from one connection to the other one represents the handover delay. Then, the probability of burst erasures \( (b_e) \) can be written as a function of both the time required by the switch (figure 4.2) to move from the link of the serving PoA to the one of the candidate PoA, and the rate of
the communication \( r \). The greater the value of \( t \) and \( r \), the greater is the burst erasure probability. Hence, according to [63], \( b_e \) can be estimated as

\[
b_e(t, r) = tr.
\]

(4.1)

On the other hand, during a soft VHO, the loss of packets are mainly due to the low level of signal between MN and PoAs. In this case, the number of erasures can also be estimated according to (4.1), since \( b_e \) is still depending on \( t \) and \( r \). In fact, the principal difference is not in the structure of the formula but in the meaning of the parameters: in fact, now time \( t \) is the time, in which the signal of the two links between MN and PoAs is so low to make the reception of frames completely ineffective. The values of \( t \) and \( r \) are very dependent on the technology and on the active services of the communication. An example of this kind of model for packet loss during VHOs can be found in [64, 65] for VHOs between IEEE 802.11n and Long Term Evolution (LTE).

Reference [63] derived a linear distribution for \( b_e \) as a function with two variables. Nevertheless, the linear model extracted from [63], is quite simple, but not very accurate to describe burst erasures due to VHOs with MIH protocol. The results presented in [27] for VHOs over MIH, provide more detailed samples to find a relatively accurate distribution for erasure probability in our Gilbert model. When the MIH protocol is deployed, the packet loss depends on three parameters [27]: number of MNs in the scenario (\( num \)), transmission rate (\( r \)) and handover time (\( t_{VHO} \)). However, the three-variable function \( b_e(num, t_{VHO}, r) \) can be converted to a two-variable function \( b_e(num, r) \), since there is a dependence of \( t_{VHO} \) on the number of MNs. So a two-variable distribution for packet loss \( b_e(num, r) \) characterises the behaviour of our channel model according to [27].

Let’s consider again the Gilbert model in (2.2) and its representation in figure 2.5. If we use the analysis extracted from [27], the matrix of probabilities can be written as

\[
\begin{bmatrix}
1 - p_{12} & p_{12} \\
1 - p_{22} & p_{22}
\end{bmatrix}
\]

(4.2)

where \( p_{12} \) is the handover probability and \( p_{22} \) is the burst erasure probability (\( b_e \)). The handover probability \( p_{12} \) depends on the scenario and on the mobility of the MNs in the network. We will use for that the values obtained for scenario in [27].
4.1. Introduction

Figure 4.4: [27, 28] Simulation scenario for VHOs between WiMAX and WLAN using the MIH protocol. 100 MNs are moving under the coverage of WiMAX base station (BS) and three WLAN access points (AP). When packet loss versus number of MNs are simulated, the transmission rate is fixed and the number of MNs varies. However, if packet loss versus rate is simulated, the number of MNs is fixed and the transmission rate of each MN changes.

In order to represent \( p_{22} = b_e \), we create the regression model of the data simulated in [27]. This model is important because provides polynomial functions, which can be used to infer the behaviour of losses during VHOs where no simulated data are available. In particular, the use of these functions is not limited to this thesis, but it can be useful for all the further works dealing with losses during VHOs with MIH. Moreover, these functions clearly summarise the relationship between \( b_e \) and the number of MNs, and \( b_e \) and the transmission rate: these dependences are important to fully characterise the performance during VHOs.

First, we show the dependence of the packet loss on the number of MNs. The scenario [27] has a WiMAX base station and three WLAN access points (figure 4.4). The MNs linearly move under the coverage of a WiMAX base station and three WLAN access points. Their speed is 50 km/h. The accuracy of this theoretical model for packet
4.1. Introduction

Figure 4.5: Burst erasure probability versus number of MNs. The transmission rate of the MNs is 1000 kb/s. The fitted curve is a polynomial of degree 5.

loss in VHOs is evaluated using the standard error\(^1\) and coefficient of determination\(^2\). The results in [27] refer to the total burst erasure probability of the system (\(b_{e,sys}\)). In particular, this value is the average \(b_e\) experienced by each transaction of each MN. Figure 4.5 shows the simulated data [27] of \(b_{e,sys}\) versus the number of MNs. The fitted curve is obtained with a polynomial of degree 4. The coefficients of the polynomial in the form \(f(x) = ax^4 + bx^3 + cx^2 + dx + g\) are: \(a = 3.906e-08\), \(b = -7.812e-06\), \(c = 0.0004219\), \(d = 0.004375\) and \(g = 0.1\). Then, its coefficient of determination is 1, and its standard error is 8.279e - 15. Moreover, figure 4.6 shows the results of simulations [27] of \(b_{e,sys}\) versus the transmission rate of each MN, when there are 100 MNs in the scenario (figure 4.4). The fitted curve is obtained with a polynomial of degree 5. The coefficients of the polynomial in the form \(f(x) = ax^5 + bx^4 + cx^3 + dx^2 + gx + h\) are: \(a = 2.014e-14\), \(b = -5.28e - 11\), \(c = 5.355e - 08\), \(d = -2.823e - 05\), \(g = 0.008814\) and \(h = -0.6087\). Then, its coefficient of determination is 0.9950300, and its standard error is 0.0223480.

\(^1\)The standard error represents the average distance that the observed values fall from the regression line. Conveniently, it tells how wrong the regression model is on average using the units of the response variable. Smaller values are better because it indicates that the observations are closer to the fitted line.

\(^2\)The coefficient of determination indicates how well data fit a statistical model.
4.2 General Architecture

Figure 4.6: Total burst erasure probability of the system versus transmission rate of MNs. The number of MNs in the network is 100. The fitted curve is a polynomial of degree 4.

Finally, after the analysis above, we can rewrite matrix (4.2) as

\[
\begin{bmatrix}
1 - 0.82 & 0.82 \\
1 - b_e & b_e
\end{bmatrix}
\]

(4.3)

where \( p_{12} = 0.82 \) is calculated from results in [27]. Depending on the variable we want to change in the model (\( r \) or \( num \)), \( b_e \) has respectively the expression of one of the polynomials previously calculated.

4.2 General Architecture

The communication model of EC-MIH maintains the same general structure of the MIH protocol, explained in figure 2.1. EC-MIH entities are still sorted into MIH MNs, MIH PoSs (PoAs and non-PoAs) and MIH non-PoSs. Moreover, MIH reference points described in subsection 2.1.1 also remain the same in the proposed EC-MIH. However, the architecture and the internal functionalities of MIHF become different in EC-MIH. Furthermore, EC-MIH protocol behaves differently from MIH one, since it exploits FEC instead of ARQ.
4.2. General Architecture

Figure 4.7: EC-MIH reference model showing the two new MIHF and EC sub-layers and their interfaces.

Figure 4.7 depicts how the reference model in figure 2.2 has been modified. The original MIHF intermediate layer has been substituted by two sub-layers:

- **MIHF sub-layer** coordinates the exchange of information and commands between upper and lower layers;

- **Enhanced-coded sub-layer** (EC sub-layer) has the main aim of generating encoded MIH frames to send to remote MIH entities. This block has interfaces with upper and lower layers to get link, application and handover related information.

MIHF sub-layer communicates with upper and lower layers respectively via MIHF_SAP and MIHF_LINK_SAP. These two SAPs includes all the primitives that originally were in MIH_SAP and MIH_LINK_SAP. Next, MIHF sub-layer interacts with EC sub-layer through an internal SAP (INT_SAP): in particular, this interface is used to exchange some control information between the two sub-layers, and to pass MIH frames to EC sub-layer in order to be encoded. In parallel, the EC sub-layer has two interfaces to exchange information with upper and lower layers respectively called EC_SAP and EC_LINK_SAP. These two SAPs are used by the sub-layer to get information both from L3 protocols and link-layer technologies in order to choose the most appropriate erasure code and to adapt its characteristics according to handover and network conditions. Furthermore, EC sub-layer sends packets to other EC-MIH remote entities via EC_NET_SAP, reserved to EC-MIH protocol encoded frames.
The introduction of a novel sub-layer with additional functionalities requires the design of further SAP primitives to support and to correctly perform coding operations. These new EC-MIH SAP primitives are listed in table 4.1. The abstract media-independent interface INT_SAP supports the primitives K_Parameter_Report, K_Parameter_Request and Size_Request. The first is an event used to send $k$ to the MIHF sub-layer. The EC sub-layer sends it once the current block of $k$ packets has successfully been sent, so the new $k$ is calculated according to the received information about the VHO. The second is K_Parameter_Request: once information packets are ready to be sent at the upper layers, the MIHF sub-layer uses this command to ask the current value of $k$ to the EC sub-layer. The last primitive is an event used to set the method to obtain packets of same size at the MIHF sub-layer. In fact, if the deployed erasure code involves RLNC, it becomes necessary to have source packets of the same size.

EC_SAP is a media-independent interface. The primitives this interface supports are: L3_Send_Parameters and L2_Parameter_Report. The former is used by upper layers to send to the EC sub-layer the fundamental parameters that can be useful to guarantee reliable communication. These parameters depend on the L3 protocol (i.e. TCP, UDP, etc.), which is exploited in the current transmission. The information included in this primitive is used by EC sub-layer in the choice of the optimal erasure code and in the calculation of its redundancy. The latter is used by EC sub-layer to report important L2 handover related parameters to the upper layers. This becomes fundamental if L3 protocols have characteristics that can be affected by variation of VHO conditions (i.e. congestion control for TCP).

Finally, EC_LINK_SAP is a media-dependent interface. The primitives it supports are: L2_Connection_Status, L2_Handover_Status and L2_Handover_Parameters. The first is used by layer 2 to provide the connection status to the EC sub-layer. This information is important to know the current status of the link and of the connection. These aspects are influencing the packet loss and are fundamental for its estimation. Moreover, this primitive is necessary to calculate the redundancy of the erasure code in order to successfully correct burst erasures. The second is exploited by L2 to transmit the handover status to the EC sub-layer. The current VHO characteristics (i.e. VHO time) are important to calculate the required redundancy to guarantee a reliable communication.
Table 4.1: INT_SAP, EC_SAP and EC_LINK_SAP primitives.

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>K_Parameter_Report</td>
<td>INT_SAP Event</td>
<td>It is used to send $k$ to the MIHF sub-layer</td>
</tr>
<tr>
<td>K_Parameter_Request</td>
<td>INT_SAP Command</td>
<td>It is used to ask the value of $k$ to the EC sub-layer</td>
</tr>
<tr>
<td>Size_Request</td>
<td>INT_SAP Event</td>
<td>It is used to set the method to obtain $k$ packets of same size at MIHF sub-layer</td>
</tr>
<tr>
<td>L3_Send_Parameters</td>
<td>EC_SAP Command</td>
<td>It is used by upper layers to send to the EC sub-layer important parameters for the code</td>
</tr>
<tr>
<td>L2_Parameter_Report</td>
<td>EC_SAP Event</td>
<td>It is used by EC sub-layer to report important parameters for the upper layers</td>
</tr>
<tr>
<td>L2_Connection_Status</td>
<td>EC_LINK_SAP Event</td>
<td>It is used by L2 to pass the connection status to the EC sub-layer</td>
</tr>
<tr>
<td>L2_Handover_Status</td>
<td>EC_LINK_SAP Event</td>
<td>It is used by L2 to communicate the handover status to the EC sub-layer</td>
</tr>
<tr>
<td>L2_Handover_Parameters</td>
<td>EC_LINK_SAP Event</td>
<td>It is used by L2 to give VHO parameters to the EC sub-layer</td>
</tr>
</tbody>
</table>

The third provides VHO parameters (i.e. link technology, authentication method employed) to the EC sub-layer to estimate the burst erasures due to handover operations. In fact, the available transmission rate affects erasure probability as the delay due to authentication procedures.

Figure 4.8 depicts the logical scheme of the EC-MIH protocol and how the EC sub-layer works. It is useful to compare this scheme with the legacy MIH protocol (figure 2.3). The first aspect that clearly appears is the lack of acknowledgement service. In fact, the MIH scheme has two blocks, called ACK-requestor and/or ACK-responder, at
both source and destination. In case of EC-MIH protocol, there are two new additional blocks that substitute those of MIH: encoder and decoder. These blocks are the ones that guarantee the reliability of the communication by using erasure-correcting codes instead of acknowledgement services. In the original MIH protocol, the acknowledgement block is running in parallel to the transaction source/destination ones. In EC-MIH protocol, the transaction source/destination blocks and the encoder/decoder blocks are working in a row. The encoder block is logically connected with two novel blocks, called timer and external information. These blocks are the responsible entities to get the information from the upper and lower layers, in order to optimise the redundancy and to achieve reliability. In particular, the timer sets the frequency at which the external information block updates the connection and VHO information. Moreover, they are logically connected with the encoder, which exploits them to decide what erasure code and how much redundancy to use. The study and design of algorithms that can be employed in the timer block and in the external information block are out of the scope of our work. However, packet loss estimation is a branch of research that has already provided several meaningful results. Briefly, the methods can be divided in active and passive [66, 67, 68, 69]: active methods estimate packet loss through live measurements, while passive methods use models of variable accuracy and sampled data in different conditions. Since the objective is to provide seamless handovers to mobile users, and to minimise latencies, we suggest the use of passive estimation methods. In fact, they may be less accurate in highly changing environments than active methods, but they are less intrusive and minimise the delay of the transmission.

Figure 4.8(b) shows how the Enhanced-Coded MIHF (EC-MIHF) acts. Once a transaction has been set up, MIHF and EC sub-layers become active. Especially, the EC sub-layer gets information from the lower layers about VHO conditions and characteristics, and from the upper layers about the protocol the communication is going to use. So, the EC sub-layer estimates the quantity of redundancy and decides the erasure code to deploy. Moreover, if needed, this layer chooses the method to obtain packets of the same size. Then, it sends the $k$ value to the MIHF sub-layer, which subsequently passes $k$ data packets to the EC sub-layer. Hence, according to the collected information, the EC sub-layer encodes the source packets with a coding rate $R = k/n$. It is important to
Figure 4.8: Logic schemes that shows the structure of EC-MIH protocol and EC sub-layer (a) Logic scheme of EC-MIH protocol (b) Internal structure of MIHF and EC sub-layer.

highlight that the redundancy must guarantee a reliable transport channel (packet loss < 0.01 as defined in [2]). At the MIHF sub-layer, a queue entity collects data frames to generate MIH ones. Then, it passes $k$ of them per time to the encoder inside the EC sub-layer. In order to optimise the performance, and to reduce the latency, it is important the design of the queue block and its relations to both the upper layers and the EC sub-layer. Nevertheless, this topic is out of the scope of this thesis. Finally, at EC sub-layer, the encoder encodes the matrix $U$ of $k$ information packets by using a defined coding scheme, and it generates the matrix $X$ of $n$ encoded packets, which are
sent to the lower layers (for example MAC layer).

4.3 Packet Structure

After previous discussion on EC-MIH reference model and on how to perform its coding operations, this subsection describes the packet structure of the EC-MIH protocol.

There are two kinds of information messages, which arrives at the EC sub-layer: variable length or fixed length. The former is possible if the coding operations do not involve RLNC, as previously described for MIH protocol. The latter is received if RLNC is applied, and the MIHF sub-layer has already produced frames of same size for the encoder. In this case, once the encoder asks the MIHF buffer for $k$ source packets, three possible techniques can be performed: first, the MIHF queue block introduces zero symbols to achieve a fixed size packet. The reference to perform this action is the longest packet in the buffer; second, the MIHF queue block may take advantage of the fragmentation mechanism; an alternative third option, which represents a trade-off between the previous ones, could be a hybrid solution, mixing the use of dummy zero symbols and fragmentation.

Reference [2] claims that an MIH message is fragmented only when MIH message is sent natively over an L2 medium. A message is fragmented when the message size exceeds a fragmentation threshold. The size of each fragment is the same except the last one, which may be smaller. The maximum fragment size is defined as the maximum value of the fragmentation threshold, which shall be equal to the Maximum Transmission Unit (MTU) of the link layer that is on the path between two MIHF nodes. When the
MTU of the link layer between two MIHF nodes is known, the maximum fragment size is set to the MTU. The method of determining such an MTU is outside the scope of the standard. When the MTU of the link layer between two MIHF nodes is unknown, the maximum fragment size is set to the minimum MTU of 1500 octets. When MIH message is sent using an L3 or higher layer transport, L3 takes care of any fragmentation issue, and the MIH protocol does not handle fragmentation in such cases. The MIH protocol header, the source MIHF identifier TLV and destination MIHF identifier TLV of the original message are copied to each fragment. However the 'variable payload length', 'more fragment', and 'fragment number' fields are updated accordingly for each fragment.

When the queue contains frames of different sizes, EC-MIH protocol decides to perform fragmentation. In this case, the EC-MIH messages are only fragmented at the EC-MIHF layer and not as in the original MIH, the layer that is in charge of sending the messages. The reference for the fragmentation is the smaller frame. If the queue uses both dummy symbols and fragmentation, an optimal size is chosen in order to balance the number of additional symbols and the amount of packets. At the destination, the MIHF sub-layer reassembles the decoded received fragments into the native frames.

Figure 4.9 depicts the structure of an encoded packet. The encoded packet’s payload is the encoded frame and contains the MIHF payload, together with source and destination MIHF identifier. Furthermore, a new header is generated for each of the \( n \) encoded packets. This header has different structure if compared to the original MIH protocol (figure 2.4).

The header of an encoded frame assists towards providing routing information to the destination. There are two possible structures, depending on the exploitation or not of RLNC. Figure 4.10(a) shows the structure of the header of EC-MIH frames when no RLNC operation is performed at the EC sub-layer, while figure 4.10(b) depicts the header of EC-MIH protocol that uses RLNC. In both cases, the 'S' field consists in two bits to identify the three possible strategies to obtain constant payload length for all the frames at MIHF sub-layer. In particular, the three cases are

00 insertion of zero dummy symbols at the end of the payload;
4.4. Overhead Analysis

When a protocol uses RLNC for error correction, there is an overhead problem to take into account. For decoding purpose, the random coefficients of the linear combinations need to be appended to the header of the packets transmitted over the links. Moreover, the overhead also includes the redundant packets sent to guarantee reliable communication over an erroneous channel.

Let’s consider a system that deploys systematic RLNC. The size of the overhead we calculate is considered per generation. The total size of the overhead due to random coefficients is $n \cdot (n - k)$, where $n$ is the total number of packets and $n - k$ is the number
4.5 Performance of the Protocol

Figure 4.11: Single-source unicast communication of an MIH transaction. This is correctly represented by a line network, which becomes the topology of simulation in ns2 to compare the behaviour of MIH and EC-MIH protocol. This line network models the behaviour of transactions during VHOs, considering losses on the wireless link according to \( b_{\text{num}} \) and \( b_{\text{e}}(r) \).

of redundant packets. Let’s consider packets of \( l \) symbols in \( \mathbb{F}_q \), with \( q = 2^m \). So, the total overhead in symbols, weighted on the total number of symbols to send in a generation, becomes

\[
\frac{(n+l) \cdot (n-k)}{n \cdot l}.
\] (4.4)

If we send packets of 1024 bits, the packet size becomes \( l = 128 \) symbols in \( \mathbb{F}_{256} \). If the generation size is \( k = 32 \) and \( n = 40 \) (\( R = 0.8 \)), the total overhead is 0.26.

In the case of EC-MIH protocol using HSCNCs, the redundancy of the outer code has to be added to this overhead. If an outer code adds 7 redundant packets to output 32 encoded ones (\( R = 0.78 \)), the result of equation (4.4) becomes 0.49.

The way to minimise the total overhead clearly appears by looking at equation (4.4): in particular, the objective is to reduce the redundancy (maximise the coding rate \( R \)), while sending packets of greater size. For example, if we send frames of 3 kB (\( l = 384 \) symbols in \( \mathbb{F}_{256} \)) the overhead of using RLNC reduces to 0.22 and the one of using HSCNCs reduce to 0.41.

### 4.5 Performance of the Protocol

In this section, we analyse the performance of EC-MIH protocol, and we compare it with the original MIH protocol. The network layer deploys TCP Reno. The performance of the two protocols is calculated in terms of average throughput achieved at the network
layer by TCP and VHO time. The software to perform the simulations is the network simulator ns2 [70]. Especially, the simulator uses two kinds of inputs:

- the range of values of $b_e$, obtained from results of simulations of MIH protocol in a particular scenario (figure 4.4)\(^3\). The values of $b_e$ are obtained depending either on the transmission rate of the MNs or on the number of MNs in the network;

- the behaviour of EC sub-layer and the coding rates required for reliable transmissions are calculated using MATLAB simulations for HSCNCs.

Finally, according to these inputs, the network simulator calculates how VHOs and coding operations affect the throughput of TCP. Once we have obtained the throughput, we can analytically calculate the handover time of both MIH and EC-MIH. Reference [28] distinguishes between two possible MIH implementations: the one with and without an MIIS server. Since the simulations of [28] reveal that the deployment of the MIIS server significantly reduces VHO time, we borrow this approach in our simulations. The main aim of MIIS server is to efficiently provide essential information for network selection.

The study and the considerations in chapter 3, showed that the use of HSCNCs can be effective and efficient to recover burst erasures. According to those results, we conclude that the HSCNCs, which are more suitable to handle losses in the considered VHO scenario, are the ones that have LT outer codes ($\delta = 4$ and $c = 0.1$), RS outer codes, and Raptor outer codes (with RS pre-coding stage).

First, let’s start by calculating the Gilbert model according to the detailed simulations of [28]. So, we derive the distribution of $b_e(num)$ and $b_e(r)$ for scenario in figure 4.4, but when the MIIS server is deployed. Furthermore, the range of transmission rates is $1 \leq r \leq 20$ Mb/s.

Figure 4.12 shows the simulated data [28] of $b_{e,sys}$ versus the number of MNs ($r = 19$ Mb/s). The fitted curve is obtained with a polynomial of degree 4. The coefficients of the polynomial in the form $f(x) = ax^4 + bx^3 + cx^2 + dx + g$ are: $a = -1.996e - 08$,

\(^3\)We want to thank the FP7 HURRICANE project [28] that has provided the tool to generate the baseline MIH performance.
4.5. Performance of the Protocol

Figure 4.12: Burst erasure probability versus number of MNs for scenario in figure 4.4. The MIH protocol takes advantage of MIIS server and the transmission rate at the application is $r = 19$ Mb/s. The fitted curve is a polynomial of degree 4.

Figure 4.13: Burst erasure probability versus transmission rate for scenario in figure 4.4. The MIH protocol takes advantage of MIIS server. The number of MNs is 55. The fitted curve is a polynomial of degree 5.
4.5. Performance of the Protocol

\[ b = 4.203e - 06, \quad c = -0.0002497, \quad d = 0.007563 \text{ and } g = 0.09083. \]

Then, its coefficient of determination is 0.9868969, and its standard error is 0.0177375. Moreover, figure 4.13 shows the results of simulations [28] of \( b_{e.sys} \) versus the transmission rate of each MN, when there are 55 MNs in the scenario of figure 4.4. Now, the variable is the transmission rate \( r \) of each MN. The fitted curve is obtained with a polynomial of degree 5. The coefficients of the polynomial in the form \( f(x) = ax^5 + bx^4 + cx^3 + dx^2 + gx + h \) are: \( a = 8.627e - 07, \quad b = -3.976e - 05, \quad c = 0.0006245, \quad d = -0.003559, \quad g = 0.01712 \) and \( h = -0.004257 \). Then, its coefficient of determination is 0.9962, and its standard error is 0.0065295.

After the analysis above, we can again write matrix (4.3)

\[
\begin{bmatrix}
1 & -0.82 & 0.82 \\
1 - b_e & b_e
\end{bmatrix}
\] (4.5)

where \( p_{12} = 0.82 \) is the handover probability obtained by simulations in [28], and \( b_e \) (depending on \( r \) or \( num \)) is the burst erasure probability, that has respectively the expression of one of the two polynomials just calculated.

The MIH topology of transactions is a single-source unicast communication, which can correctly be represented by a line network, in which a source sends packets to a sink. Moreover, an MIH transaction can involve some intermediate nodes (see figure 4.11). The losses we are interested in this context, are the ones only referred to VHOs, which can be placed on the wireless link. So, the simulations consider a source, which runs a File Transfer Protocol (FTP) application, connected to a sink. It emits packets continuously from the start time until the end time. The application uses TCP Reno with congestion control. The TCP source transmits at 20 Mb/s. The wireless links exploit technologies such as IEEE 802.11n and WiMAX, which can guarantee to support the constant transmission rate of the application. The propagation delay of wireless links is 1 ms. The wired link, which connects the PoA to the sink, has bandwidth of 1 Gb/s and propagation delay of 5 ms. The packet size is fixed at 1 kB. The buffer size at the intermediate and sink node is set at 100 packets, and the window size of TCP is set at 50 packets. According to the considerations above, the Gilbert error model is only active on the wireless links. In these simulations, intermediate nodes do not re-encode
4.5. Performance of the Protocol

packets since wired links are perfect channels. This also minimises the transmission and handover delay. The simulated Markov model reproduces the Gilbert model described by matrix 4.5. So, VHO probability is $p_{12} = 0.82$. Moreover, the probability of ending successfully a VHO procedure is 0.94. The simulation time is set at 100 s.

Table 4.2 summarises the main parameters of simulations that are used to obtain the results in the next sections. The average throughput is evaluated for both $b_e(num)$ and $b_e(r)$. In case of $b_e(num)$, the MN application transmits information at the same bit rate ($r = 19$ Mb/s), while the number of MNs in the network increases. On the other hand, in case of $b_e(r)$, the erasure probability affects the throughput because the MN application transmits at different rates during VHOs; the number of MNs in the network is fixed ($num = 55$). Reference [71] confirms that the transmission rates of TCP source are supported by the two involved wireless technologies (IEEE 802.11n and WiMAX). Finally, the simulation time of 100 s alternates periods of transmissions ($t_{tr}$) to periods of handover ($t_{VHO}$). The ratio between them can be set at $t_{VHO}/t_{tr} = 0.1$ [28].

Figure 4.14 depicts the instantaneous throughput versus the simulation time for MIH protocol. The application running at the MN has transmission rate $r = 19$ Mb/s. Since both technologies (IEEE 802.16m and IEEE 802.11n) support the transmission rate of the source application, the throughput is just affected by losses due to VHO procedures. The $b_e$ varies according to the number of MNs in the scenario: in particular $b_e(10) = 0.15$. The achieved average throughput of the communication during 100 s is around 16.2 Mb/s. There are four VHOs during 100 s of simulation. It is possible to see that the burst erasures due to VHOs cause the TCP instantaneous throughput to fall sharply. Especially, figure 4.14(b) shows a snapshot of a single handover ($21 \leq t \leq 26$) to clarify the effect of VHOs. Moreover, it plots the curves for $b_e(10) = 0.15$ and $b_e(50) = 0.25$ in order to describe how a variation of $b_e$ affects the throughput.

Figure 4.15 shows how the increase of handover time – compared to the transmission time – affects the throughput of the communication during the 100 s of simulation. The transmission time $t_{tr}$ is the time, in which the MN is in the state $s1$ of the Gilbert model. The scenario has 50 MNs, $b_e = 0.25$ and $r = 19$ Mb/s. The average throughput
Table 4.2: Parameters of simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>100 s</td>
</tr>
<tr>
<td>Wired link</td>
<td>4 ms</td>
</tr>
<tr>
<td></td>
<td>1 Gb/s</td>
</tr>
<tr>
<td>Delay wireless link</td>
<td>1 ms</td>
</tr>
<tr>
<td>TCP parameters</td>
<td>FTP constant bit rate</td>
</tr>
<tr>
<td></td>
<td>Window size 50</td>
</tr>
<tr>
<td></td>
<td>Packet size 1 kB</td>
</tr>
<tr>
<td>WiMAX parameters</td>
<td>IEEE 802.16m</td>
</tr>
<tr>
<td></td>
<td>Data rate 100 Mb/s</td>
</tr>
<tr>
<td>WLAN parameters</td>
<td>IEEE 802.11n</td>
</tr>
<tr>
<td></td>
<td>Data rate 300 Mb/s</td>
</tr>
<tr>
<td>Simulation with ( b_e(r) )</td>
<td>( 1 \leq r \leq 20 ) (Mb/s) [71]</td>
</tr>
<tr>
<td></td>
<td>( num = 55 )</td>
</tr>
<tr>
<td>Simulation with ( b_e(num) )</td>
<td>( 10 \leq num \leq 100 )</td>
</tr>
<tr>
<td></td>
<td>( r = 19 ) Mb/s [71]</td>
</tr>
<tr>
<td>Handover time</td>
<td>( t_{VHO} = 0.1 \cdot t_{tr} ) [28]</td>
</tr>
</tbody>
</table>
4.5. Performance of the Protocol

Figure 4.14: Throughput versus time for MIH protocol (a) Throughput achieved by a MN with transmission rate \( r = 19 \) Mb/s versus total simulation time, for \( b_e(num) \) with 10 MNs \( (b_e(10) = 0.15) \). The MN performs four VHOs (b) Snapshot of the throughput during the VHO in the time range \( 21 \leq t \leq 26 \). The comparison of \( b_e(10) = 0.15 \) and \( b_e(50) = 0.25 \) mainly shows how throughput is lower by increasing \( b_e(num) \).
4.5. Performance of the Protocol

Figure 4.15: Throughput versus time for MIH protocol. The scenario has 50 MNs, $b_e = 0.25$ and transmission rate $r = 19$ Mb/s (a) Throughput achieved by a MN versus simulation time, for $t_{VHO} = 0.1 \cdot t_{tr}$. The MN performs four VHOs (b) Snapshot of the throughput during a VHO in the time range $20 \leq t \leq 35$. The comparison of the two curves referred to $t_{VHO} = 0.1 \cdot t_{tr}$ and $t_{VHO} = 0.4 \cdot t_{tr}$ shows the impact of VHO time on throughput.
4.5. Performance of the Protocol

Figure 4.16: Average throughput versus coding rate of EC-MIH protocol HSCNCs when $r = 19$ Mb/s.

in 100 s for $t_{VHO}/t_{tr} = 0.1$ is 16.2037. Figure 4.15(b) depicts a snapshot of throughput in the range $20 \leq t \leq 35$. Furthermore, it plots throughput by increasing VHO time ($t_{VHO}/t_{tr} = 0.1$ and $t_{VHO}/t_{tr} = 0.4$). Thus it confirms the importance of minimising the handover time and highlights the importance of deploying MIIS servers [28] during VHOs.

Finally, figure 4.16 shows how the average throughput of EC-MIH protocol decreases during an handover procedure (i.e. the VHO considered is the one in figure 4.14(b)), when the coding rate decreases from 1 to 0.5. The horizontal axis represents the coding rate required by the protocol to guarantee reliable communication, when the VHO causes losses according to $b_e(num)$. The transmission rate is $r = 19$ Mb/s.

4.5.1 Luby Transform Outer Codes

Let’s consider EC-MIH protocol which uses the HSCNC constituted by LT outer codes ($\delta = 4$ and $c = 0.1$) and inner systematic RLNC. The protocol changes the coding
Figure 4.17: Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with LT outer codes ($\delta = 4$ and $c = 0.1$). The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq num \leq 60$. 
rate to guarantee the reliability of the transmission at each value of $b_e$. As previously discussed, $b_e$ varies according to the number of MNs in the network and according to the transmission rate $r$. The coding operations at EC sub-layer introduce a delay $t_{enc}$ at the source, and $t_{dec}$ at the receiver. However, if the codes are implemented in an efficient way [72], the latency and the limitation due to coding operations can be negligible.

Figure 4.17 depicts the average throughput of MIH and EC-MIH protocol during VHO with $b_e(num)$. The erasure probability changes because the number of MNs increases from 1 to 60. The transmission rate of TCP is fixed at $r = 19$ Mb/s. The erasures strongly decrease the average throughput of TCP over MIH protocol during VHOs. The deployment of these HSCNCs is capable to achieve an average throughput improvement of between 16% and 22%, when $0.01 \leq b_e(num) \leq 0.3$. Moreover, the results show that EC-MIH protocol reaches an average throughput that is close to the one of TCP (between 95% and 80%).

On the other side, figure 4.18 depicts the average throughput of MIH and EC-MIH protocol during VHO with $b_e(r)$. This time the erasure probability changes because the transmission rate of TCP increases. Even in this case, the average throughput improvement is clear and the EC-MIH protocol outperforms the one of MIH. The average throughput gain varies between 9% and 29%. The results display the same behaviour as for $b_e(num)$: EC-MIH protocol reaches an average throughput that is between 0.96 and 0.8 of the one of TCP.

Figure 4.19 compares the performance of MIH and EC-MIH in terms of handover time. In particular, the results shows the time needed by both protocols to complete a handover that requires the transmission of 250 kB. By comparing figure 4.19 and figure 4.17, it appears that the reduction of VHO time is directly proportional to the average throughput gain of EC-MIH protocol.

Finally, let’s analyse the overhead these HSCNCs require. In order to do that, we use expression (4.4). The coding rate to guarantee reliable communication is $0.9 \leq R \leq 0.6$. This means that the overhead of EC-MIH protocol varies from 0.04 to 0.44.
4.5. Performance of the Protocol

Figure 4.18: Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with LT outer codes ($\delta = 4$ and $c = 0.1$). The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq r \leq 20$ Mb/s.
4.5. Performance of the Protocol

Figure 4.19: Comparison of vertical handover time of MIH and EC-MIH protocol with HSCNCs with LT outer codes ($\delta = 4$ and $c = 0.1$). The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq \text{num} \leq 60$. The size of source information is $250$ kB.

4.5.2 Reed-Solomon Outer Codes

The HSCNCs, constituted by RS outer code and inner systematic RLNC are used to achieve a reliable communication. According to the evaluation in chapter 3 and the results in [72], the values of $t_{\text{enc}}$ and $t_{\text{dec}}$ can be neglected in our calculations.

Figure 4.20 represents the average throughput of MIH and EC-MIH protocol during VHO with $b_e(\text{num})$. As before, the erasure probability changes because the number of MNs increases from 1 to 60. The transmission rate of TCP is fixed at $r = 19$ Mb/s. The use of these HSCNCs gets an average throughput improvement between 19% and 30% percent, when $0.01 \leq b_e(\text{num}) \leq 0.3$. Moreover, the results show that EC-MIH protocol reaches an average throughput that is close to the one of TCP (between 98% and 83%).

Next, figure 4.21 shows the average throughput of MIH and EC-MIH protocol during VHO with $b_e(r)$. Hence, the erasure probability changes because the transmission rate of the source increases. The average throughput improvement is significant and
Figure 4.20: Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with RS outer codes. The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq num \leq 60$. 
Figure 4.21: Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with RS outer codes. The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq r \leq 20$ Mb/s.
4.5. Performance of the Protocol

Figure 4.22: Comparison of vertical handover time of MIH and EC-MIH protocol with HSCNCs with RS outer codes. The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq num \leq 60$. The size of source information is 250 kB.

EC-MIH protocol outperforms MIH one. The average throughput gain varies between 9% and 32%.

Figure 4.22 shows the performance of MIH and EC-MIH in terms of handover time. The amount of information to be sent to complete the handover is 250 kB. By comparing figure 4.22 and figure 4.20, it also appear in this case that the amount of reduction of VHO time directly follows the average throughput gain of EC-MIH protocol.

Regarding the overhead, the coding rate to guarantee reliable communication is $0.9 \leq R \leq 0.65$. This means that the overhead of EC-MIH protocol varies from 0.04 to 0.36. Then, by using RS outer codes, the maximum overhead required is 0.08 less than the one of HSCNCs with LT outer codes.

4.5.3 Raptor Outer Codes

Finally, we study the performance of EC-MIH protocol, which uses HSCNCs constituted by Raptor outer codes (with RS pre-coding) and inner systematic RLNC. As before,
Figure 4.23: Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with Raptor outer codes (RS pre-coding stage). The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq \text{num} \leq 60$. 
4.5. Performance of the Protocol

Figure 4.24: Average throughput versus burst erasure probability of EC-MIH protocol with HSCNCs with Raptor outer codes (RS pre-coding stage). The burst erasure probability changes in \( 0.01 \leq b_e \leq 0.3 \) for \( 1 \leq r \leq 20 \) Mb/s.

The value of \( t_{\text{enc}} \) and \( t_{\text{dec}} \) are considered negligible because of their marginal effect due to current very efficient real implementations [72].

Figure 4.23 displays the average throughput of MIH and EC-MIH protocol during VHO with \( b_e(\text{num}) \). The erasure probability changes because the number of MNs increases from 1 to 60. The transmission rate of TCP is fixed at \( r = 19 \) Mb/s. The deployment of HSCNCs with Raptor outer codes, achieve an average throughput improvement of between 16% and 24%, when \( 0.01 \leq b_e(\text{num}) \leq 0.3 \). Furthermore, EC-MIH protocol reaches an average throughput that is close to the one of TCP (between 92% and 88% percentage), especially when the burst erasure probability augments.

Then, figure 4.24 shows the average throughput of MIH and EC-MIH protocol during VHO with \( b_e(r) \): the erasure probability changes because the transmission rate of the source increases. The results quantify how the EC-MIH protocol outperforms the MIH
Figure 4.25: Comparison of vertical handover time of MIH and EC-MIH protocol with HSCNCs with Raptor outer codes (RS pre-coding stage). The burst erasure probability changes in $0.01 \leq b_e \leq 0.3$ for $1 \leq num \leq 60$. The size of source information is 250 kB.

one. The average throughput gain varies between 4% and 25% percent.

Figure 4.25 shows the performance of MIH and EC-MIH in terms of handover time. The source information required to complete the handover is 250 kB. The comparison between figure 4.25 and figure 4.23 reveals that the reduction of VHO time of the EC-MIH protocol is directly proportional to the average throughput gain.

This time, the coding rate to guarantee reliable communication is $0.8 \leq R \leq 0.7$. This means that the overhead of EC-MIH protocol varies from 0.25 to 0.36. The minimum overhead of this coding scheme is higher than the previous ones, since it consists of three erasure codes, serially concatenated.

### 4.6 Summary and Conclusion

The chapter addressed the problem of designing an efficient and effective EC-MIH protocol based on FEC, to outperform the performance of MIH protocol during VHOs.
4.6. Summary and Conclusion

The evaluation of the performance has been measured in terms of average throughput and average throughput gain. The results obtained with the ns2 simulator considered as input, the results previously obtained in chapter 3 and in [28]. In order to summarise the discussion of this chapter, we use figure 4.26 and figure 4.27.

Figure 4.26 shows the average throughput gain of EC-MIH protocol when the EC sub-layer uses HSCNCs with either LT outer codes, RS outer codes or Raptor outer codes (with RS pre-coding). The range of $b_e$ varies between 0.01 and 0.3 when the number of MNs in the scenario changes between 1 and 60. The circles in the figure highlight the maximum gain. The highest average throughput gain is achieved by HSCNCs with RS outer codes: its maximum is 30% at $b_e = 0.1$. Instead, the lowest average throughput gain is achieved by HSCNCs with LT outer codes: its maximum is 22% at $b_e = 0.05$. The HSCNCs that employ Raptor outer codes are the ones that roughly maintain a constant throughput gain while $b_e$ increases: it almost keep is maximum (24%) for $b_e \geq 0.1$.

Next, figure 4.27 depicts the average throughput gain of EC-MIH protocol when the EC sub-layer uses HSCNCs with either LT outer codes, RS outer codes or Raptor outer
Figure 4.27: Average throughput gain versus $b_e(r)$ of EC-MIH protocol with HSCNCs.

codes (with RS pre-coding). The range of $b_e$ varies between 0.01 and 0.31 when the source rate increases from 1 Mb/s to 20 Mb/s. The circles in the figure emphasise the achieved maximum gain. This time, the maximum average throughput is always obtained when $b_e = 0.15$. As previously in case of $b_e(num)$, the highest average throughput gain is achieved by HSCNCs with RS outer codes: its maximum is 32%. The lowest average throughput gain is attained by HSCNCs with LT outer codes for $b_e > 0.05$: its maximum is 29%. Finally, the HSCNCs with Raptor outer codes have a maximum average throughput gain of 25%.

According to the results described in this chapter, the EC-MIH protocol can guarantee a reduction of the handover time that equals the average throughput gain.

Regarding the analysis of the overhead, HSCNCs with RS outer codes are the most efficient codes. They achieve the highest erasure correction for less overhead than the other two schemes. Nevertheless, Raptor outer codes (with RS pre-coding) represent an efficient solution as well, especially at high burst erasure probabilities (i.e. $b_e > 0.25$).

According to the results of this chapter, we can claim that EC-MIH protocol with HSCNCs always outperforms MIH in terms of average throughput achieved and VHO
time. Among the coding schemes analysed, we suggest the implementation of HSCNCs with RS outer codes, since they are the most effective.
Chapter 5

Conclusion

This final chapter overviews and highlights the significant results of this thesis. Moreover, we suggest some possible directions of research in order to continue and further improve the achievements of our work.

5.1 Main achievements

A principal characteristic of future 5G mobile networks will be the user capability to seamlessly move between different wireless technologies. An efficient technology-independent solution to provide seamless VHOs was released by IEEE in 2008. Beyond a specific architecture, the standard IEEE 802.21 describes a so called MIH protocol, which is responsible to exchange handover control messages among network entities, and to guarantee reliable communications (packet loss less than 0.01) during the handover. In particular, the reliability is provided by exploiting ARQ. However, the acknowledgement service and the retransmission policies can cause a rapid fall of the communication throughput in highly lossy channels. This significant drawback led us to choose an alternative method to control the reliability of MIH transactions: channel coding.

FEC is an efficient technique to control erroneous communications by adding redundant information. This redundant information has to be optimised to keep the consumption
of extra bandwidth low. Therefore, this provided the impetus for an integrated design that could combine FEC with MIH to provide enhanced link resiliency. However, this raised open challenges in the code design that requires a large minimum distance and needs to be low complexity in nature. A good approach to address this task considered the design of concatenated coding schemes, that has widely been used in previous works to enhance the transparency of the link to channel erasures. However, concatenated codes can be highly complex. In order not to increase the complexity of these schemes, first we limited the concatenation to two codes.

In the last decade, the importance of RLNC in error-correction increased. NEC codes were demonstrated to be very good codes for erasure recovery. However, as we described, these coding schemes have a relevant drawback: the minimum distance is a random variable.

Our novel approach aimed to merge the positive characteristics of classical erasure codes and NEC codes to design powerful concatenated codes for VHOs, called HSCNCs. Furthermore, we kept the complexity low to make them applicable. HSCNCs exploit both rateless RLNC as inner code and outer erasure codes to mitigate the effect of the randomness of RLNC.

The design of a powerful coding scheme requires the detailed study of the error pattern. Hence, we collected simulated data of packet loss of MIH during VHOs. Especially, these data showed the dependence of frame loss on transmission rate and number of mobile nodes in the network. This input allowed us to develop a numerical model of losses during VHOs by providing polynomial functions, which approximate the simulated behaviour: in this way it is possible to correctly study the loss pattern. The structure of the loss pattern helped to design the HSCNCs towards burst-erasure correction in the new EC-MIH protocol.

This thesis went beyond legacy MIH by incorporating FEC services to enhance the performance during VHO events. In this context, we propose the so called EC-MIH protocol. The EC-MIH protocol represents a new architecture, and logical scheme to complement the general MIH approach.

The first objective of this thesis was the design of an efficient and effective network
coding based solution for FEC, that constitutes a fundamental building block of the EC-MIH solution. We decided to consider as outer codes of HSCNCs, the most common erasure codes such as LT codes, RS codes (in particular FRS codes), LDPC codes, and Raptor codes (with LDPC pre-coding). Moreover, we implemented and evaluated a 'somewhat’ novel Raptor code exploiting RS pre-coding.

The evaluation of the erasure-correcting capabilities of the codes was obtained with MATLAB implementation and simulations, supported by some theoretical work. We found that some HSCNCs can outperform the error correction capabilities of RLNC. The most efficient and effective coding schemes were the HSCNCs either employing RS outer codes or with Raptor code (with RS pre-coding). By considering efficiency as a primary constraint, HSCNCs with LT outer codes can be chosen, even if they require slightly lower coding rates. The codes using LDPC resulted to be not so flexible and required coding rates less than 0.5 (not efficient for VHOs).

The second objective of this thesis was the detailed design of a novel protocol to perform seamless VHOs. This protocol, called EC-MIH, mainly aimed to outperform the throughput performance of MIH protocol. The EC-MIH protocol masks the loss of frames with the deployment of erasure codes: in particular, it uses the new proposed HSCNCs. The novel protocol guarantees reliable and more efficient VHO procedures by providing seamless handovers to mobile users. The use of erasure codes to improve mobile experience was justified by the possibility to model packet loss during VHOs as a burst erasure channel. Then, we designed a Gilbert model with particular error distributions to accurately describe the erasures during handovers with MIH.

The simulation results to obtain the average throughput of MIH and EC-MIH protocol were obtained with the network simulator ns2. Especially, the ns2 simulator takes advantage of the measurements of MIH packet loss calculated with the simulator [28]. Our results led to the conclusion that EC-MIH protocol with HSCNCs significantly outperformed MIH performance. We showed that the burst erasure probability varies according to the number of nodes in the network and to the transmission rate at the source. In both cases, the improvement in terms of average throughput could surpass the 30% level.
The range of $b_e$ changes between 0.01 and 0.3 when the number of MNs is between 1 and 60. The HSCNC which achieved the highest average throughput is the one with RS outer codes: in fact, its maximum is 30%. On the other hand, the lowest average throughput gain is attained by HSCNC with LT outer codes: its maximum is 22%. Nevertheless, if we consider the transmission rate, the burst erasure probability varies between 0.01 and 0.31 when the source rate increases from 1 Mb/s to 20 Mb/s. The highest average throughput gain is still achieved by HSCNCs with RS outer codes: its maximum is 32%. The lowest average throughput gain was achieved by HSCNCs employing LT outer codes: its maximum is 29%. The novel HSCNC with Raptor outer codes (RS pre-coding) resulted to be the most effective and efficient for high burst erasure probability ($b_e > 0.25$).

In our throughput simulations we also considered the effect of overhead due to redundant information and coefficients of linear combinations of RLNC. The achieved average throughput with EC-MIH protocol showed that the effect of this overhead slightly limited the improvements.

Moreover, the results also revealed that the average throughput gain guarantees to EC-MIH a reduction of about the same percentage in VHO time. In fact, HSCNC with LT outer codes achieved the maximum reduction (24%) for $b_e = 0.05$ and $b_e = 0.1$. HSCNCs with RS outer codes provided a reduction in handover time of maximum 32% for $b_e = 0.2$ and $b_e = 0.25$. Next, HSCNC with Raptor outer codes (RS pre-coding) resulted in a reduction of maximum 28% for $b_e \geq 0.2$.

5.2 Future work

The work in this thesis touches on several different research disciplines. The achieved results range from coding theory and network coding, to analytical modelling of VHOs, and IEEE 802.21. In this section we are going to identify potential research directions on these areas, as inspiration for future work in this area.

- The concatenation of RS codes and RLNC opens the research door on efficient decoding procedures for these hybrid approaches. Since the operations of RS
codes can be expressed in matrix form (see [73]), it may be possible to design fast
decoders for HSCNCs based on matrix operations.

- If we use HSCNCs with LT outer codes, it is important to find new different
distributions for the generator matrix, which are more efficient and effective when
$k \cdot l < 10^6$. Moreover, it is interesting to investigate the application of systematic
LT codes.

- HSCNCs with LDPC outer codes and inner RLNC requires more investigation.
In particular, it is necessary to design more flexible and more powerful burst-
erasure LDPC codes. Moreover, it is possible to design more efficient decoding
algorithms for these HSCNCs.

- The concatenation of Raptor codes and RLNC also requires more investigation.
Especially, it would be significant to study the impact of different pre-coding
schemes for Raptor codes in order to optimise the performance with RLNC.
Moreover, the RS pre-coding needs further research: it should be possible to
reduce the complexity of this Raptor coding scheme by designing a novel fast
decoding algorithm.

- The outer codes for HSCNCs proposed in this thesis are the most famous and
common for burst-erasure correction. However, it should be interesting to look
for other erasure-coding schemes to exploit as outer codes, which can achieve high
performance if concatenated with RLNC.

- Another possibility for further study is the design of concatenated codes that
exploit Subspace inner codes. Especially, it could be useful to start with Subspace
codes developed in [74], and to analyse their performance with LT, RS and Raptor
outer codes.

- The analytical modelling of VHOs with MIH is a very new field of research. In
order to design accurate distributions for $b_e$ to include in the Gilbert model,
it can be fundamental to deeply study the behaviour of packet loss in several
different scenarios, with different technologies. Furthermore, it may be possible
to find other models for channel with memory, which add some complexity but also increase the accuracy of VHO modelling.

- The design of the new EC-MIH protocol opens a new field of research. It may be important to know the performance in terms of throughput with UDP and SCTP. Moreover, we suggest the simulation of the protocol with specific technologies at physical layer to understand the interaction with different physical layer standards. Finally, it could be interesting to enhance EC-MIH by considering the authentication issues faced by IEEE 802.21a amendment.

- There is a very recent field of research that is focused on multi-hop MIH. This new version of MIH protocol could be enhanced by exploiting EC-MIH protocol. In fact, the use of FEC and especially RLNC can be good, not only for error correction, but also to improve the throughput and the efficiency of the communications.

- Finally, HSCNCs can be easily applied to enhance the performance in all the scenarios, in which signalling or data transmissions experience burst erasures.
Bibliography


[7] I. Reed and G. Solomon, “Polynomial codes over certain finite fields,” Journal of


[28] 7th Framework Programme, “Handovers for ubiquitous and optimal broadband connectivity among cooperative networking environments (FP7 HURRICANE).”


