The quality of data services over digital broadcast radio

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:


Metadata Record: https://dspace.lboro.ac.uk/2134/35349

Publisher: © John Nicholas Whitley

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
Please note that fines are charged on ALL overdue items.
The Quality of Data Services Over Digital Broadcast Radio

by

John Nicholas Whitley

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy of

Loughborough University

16th May 2008

Copyright 2008 John Nicholas Whitley
Abstract

There are a number of competing digital broadcast radio standards in existence today. These were primarily developed to provide a number of either audio only, or audio-visual services across a substantial geographical area. Since their conception, all of the digital broadcast standards have been able to transmit other kinds of data, known as data services, alongside audio and audio-visual data.

The ability of audio and video codecs to cope with reception affected by error tends to be much higher than other kinds of digital data. As these standards were primarily developed to be carriers of audio or audio-visual data, the error protection for all of the digital broadcast radio standards was designed to be suitable for the contemporary audio-visual codecs; typically the worst expected Bit Error Rate (BER), after the error protection, is $10^{-4}$. This protection is not strong enough for data that is intolerant of error.

This research analyses and compares the performance of data services sent on Digital Audio Broadcast (DAB) and Digital Video Broadcast - Hand-held (DVB-h) channels. To achieve this comparison a network metric, dubbed channel performance, was developed, which combines the efficiency of a transmission with the quality of reception. This provides a mechanism to compare standards unaffected by the bandwidth available to the channel, allowing a direct comparison between configurations of standards which would otherwise be hard to make.

Also in this research many models and a broadcast digital radio simulator were developed. The models work with the assumption of a Uniform Random BER. The simulator can be given many different error profiles. As part of this research, real data was captured and analysed. The results of this real data, along with the results of the experiments executed upon the models, are presented. The results of these provide a new understanding of how best to configure a digital broadcast radio network, and a new comparison of the available standards.

Keywords: Digital Broadcast Radio, Network Model, Network Simulation, Channel Performance, DAB, DMB, DVB-h, Data Services
Acknowledgements

I owe a great deal to the kindness and support of my supervisors Dr Iain W. Phillips and Dr Ana Sälägean, and to my director of research Dr Chris Hinde. They have been a valuable source of knowledge, and a much needed support in the hard times. And thanks for all of the coffee.

My thanks to both Dr Frank Ball and Prof. David Parish for reading this work, and taking me though my viva. It’s very kind of you to do so much reading.

I owe a debt of thanks to Mark Sutcliffe and Simon Mason who supported and encouraged me though my research, and were always interested to hear of any discoveries that I made. Thank you for your time and effort.

There are a large number of people who have become friends, beyond those already named here, who have had a positive influence on my time as a postgraduate student: within my research group in Computer Science my particular thanks to Mark Withall and Mathew Tylee Atkinson for reading my work in the development stages, providing polite and kind corrections where I did not quite type that which I meant, and not allowing me to get too distracted with other interesting research topics. Also within my research group, I’m grateful to José Ignacio Rendo Fernández, Jose Alberto Hernandez, Lezan Hawizy, Alexandra Alecu, Andy Payne, Lin Guan; within my adopted research group in Electrical Engineering, M. Shirantha de Silva and Andy Larkum; and within Sport Science, Emma Rich. You have all helped me stay sane and given me a deeper understanding of computer science and life, as well as entertaining me with (rather too many) coffee breaks. I hope that it’s been as much fun for you as it has been for me: thank you all.

The relationships that I have been privileged to be in at Emmanuel Church, Loughborough have been pillars of strength in times of real need over the last few years, as well as being sources of near-endless entertainment. Particularly Jon and Alison Walker; Steve, Cathy, Chris and Nick Threlfall-Rogers; Pete and Rachel Sandford; Mark and Rebecca Sandford; Harley; and Pete Coombs. You have all, very differently, been of great support and kindness. Thank you all for your friendships. With out you, I doubt that this would ever have been finished.

To Ray and Bex Thomas; and to Mike, thank you for your continued care for
me. You have watched over me for so long, and I am deeply indebted to you.

I am incredibly privileged to have the family that I have: my parents, Nick and Helen; my brothers Chris and Richard. You have been so supportive over my time as a student again. Thank you so much. Without you life would have been so much more boring, and cold.

There have been two bible passages that have spoken to me particularly during the last few years; my preferred interpretation of these come from The Message translation [50]:

Go to work in the morning and stick to it until evening without watching the clock. You never know from moment to moment how your work will turn out in the end.
Ecclesiastes 11:6† [50]

Don’t fret or worry. Instead of worrying, pray. Let petitions and praises shape your worries into prayers, letting God know your concerns. Before you know it, a sense of God’s wholeness, everything coming together for good, will come and settle you down. It’s wonderful what happens when Christ displaces worry at the centre of your life.
Philippians 4:6–7† [50]

I dedicate this work to my saviour Jesus Christ, who gave me life and then taught me how to live.

# Contents

1 Introduction

1.1 Digital Radio Data Services ........................................... 1
1.2 Alternative Technologies ........................................... 2
  1.2.1 Mobile Telephony ........................................... 3
  1.2.2 Wireless Networking for Hand-held Computers .......... 3
1.3 Broadcast Versus Unicast ........................................... 4
1.4 Modelling and Simulation ........................................ 4
1.5 Contributions of this work ........................................ 5
1.6 Thesis Structure .................................................. 5

2 Background

2.1 Broadcast Information Systems .................................... 6
2.2 Literature Review .................................................. 7
  2.2.1 Overview of Existing Wireless Digital Communication Tech-
  nologies .................................................. 8
  2.2.2 Work Proceeding and Completed in the area .............. 13
  2.2.3 Performance Metrics for Digital Broadcast Networks .... 15
  2.2.4 Error Handling Techniques ................................ 16
  2.2.5 PDU Header Compression ................................... 23
2.3 Understanding the Mechanics of the Standards ................. 24
  2.3.1 DAB/DMB/DAB+ Ensemble .................................. 25
  2.3.2 MSC SPM ............................................ 26
  2.3.3 MSC EPM ............................................ 28
  2.3.4 Datagroups over either MSC SPM or MSC EPM ........... 31
  2.3.5 IP over Datagroups over either MSC SPM or MSC EPM .... 39
  2.3.6 MPEG2-TS over MSC Stream Mode ........................ 41
  2.3.7 IP over MPE-FEC over DVB-h ............................. 43
  2.3.8 ROHC ............................................... 46
2.4 Summary .......................................................... 48
3 Performance Modelling

3.1 Network Quality ........................................... 49

3.2 Modelling MSC SPM ....................................... 52
  3.2.1 Mathematical Model of Probability of Correct Receipt .. 52
  3.2.2 Mathematical Model of Time Taken ..................... 53
  3.2.3 Enhanced Mathematical Model of Probability of Correct Receipt ........................................ 54
  3.2.4 Enhanced Mathematical Model of Time Taken ............ 54

3.3 Modelling MSC Datagroups over MSC SPM ................... 54
  3.3.1 Mathematical Model of Probability of Correct Receipt .. 55
  3.3.2 Mathematical Model of Time Taken ..................... 56
  3.3.3 Mathematical Model To Find The Minimum Number Of Carousel Rotations Required ..................... 56
  3.3.4 Enhanced Mathematical Model of Probability of Correct Receipt ........................................ 57
  3.3.5 Enhanced Mathematical Model of Time Taken ............ 58

3.4 Modelling MSC EPM ......................................... 58
  3.4.1 Mathematical Model of Probability of Correct Receipt .. 58
  3.4.2 Mathematical Model of Time Taken ..................... 59
  3.4.3 Enhanced Mathematical Model of Probability of Correct Receipt ........................................ 60

3.5 Modelling MSC Datagroups over MSC EPM ................... 60
  3.5.1 Mathematical Model of Probability of Correct Receipt .. 61
  3.5.2 Mathematical Model of Time Taken ..................... 62
  3.5.3 Enhanced Mathematical Model of Probability of Correct Receipt ........................................ 62

3.6 Modelling MPEG2-TS over MSC Stream Mode ................. 62
  3.6.1 Mathematical Model of Probability of Correct Receipt .. 63
  3.6.2 Mathematical Model of Time Taken ..................... 64

3.7 Modelling IP over MPE-FEC over DVB-h ..................... 64
  3.7.1 Mathematical Model of Probability of Correct Receipt .. 64
  3.7.2 Mathematical Model of Time Taken ..................... 66

3.8 Software Simulation of the Standards ....................... 66
  3.8.1 Design of the Software Simulator ...................... 66
  3.8.2 Design of the System ................................ 74

3.9 Object size Analysis ..................................... 75
  3.9.1 MPEG Streams ....................................... 77

3.10 Error Profiling .......................................... 77
4 Evaluation and Results

4.1 Performance Metric ............................................. 80

4.1.1 Performance of a reliable channel, including transmission of overhead ............................................. 81

4.1.2 Performance of a lossy channel with reliable reception after two transmissions, including transmission of overhead ............................................. 82

4.1.3 Performance of a lossy channel after two transmissions, including transmission of overhead ............................................. 82

4.1.4 Performance of a lossy channel needing increasing transmissions for reliable reception, including transmission of overhead ............................................. 83

4.2 Employing the Mathematical Model .................................. 84

4.2.1 MSC SPM .................................................................... 85

4.2.2 MSC EPM .................................................................... 91

4.2.3 Datagroups over MSC SPM ........................................... 96

4.2.4 Datagroups over MSC EPM ........................................... 105

4.2.5 The comparison of the TDCs ......................................... 108

4.2.6 IP/MSC Datagroups/MSC EPM for Streamed Data .................. 114

4.2.7 IP/MSC Datagroups/MSC SPM for Streamed Data .................. 119

4.2.8 MPEG2-TS/MSC Stream Mode .................................... 120

4.2.9 Streamed Data over IP/MPE-FEC .................................... 125

4.2.10 The Comparison of the Stream Data Standards .................... 129

4.2.11 IP/MSC Datagroups/MSC SPM for Cached Data .................. 135

4.2.12 IP/MSC Datagroups/MSC EPM for Cached Data .................. 139

4.2.13 MPE-FEC over DVB-h .............................................. 140

4.2.14 Comparing the Standards ............................................ 143

4.3 Enhanced Mathematical Model ......................................... 149

4.3.1 Comparison of the performance of the MSC Datagroups/MSC SPM models ............................................. 149

4.3.2 Comparison of the models of the RS FEC mechanisms .................. 155

4.4 Software Simulation .................................................... 156

4.4.1 Validating the Simulator .............................................. 157

4.4.2 Streamed data over IP over MSC Datagroups over MSC SPM160

4.4.3 Streamed data over IP over MSC Datagroups over MSC EPM165

4.4.4 Streamed Data over MPEG-2 TS over MSC Stream Mode .... 169

4.4.5 Streamed Data over IP over MPE-FEC ......................... 172

4.4.6 Comparison of Streamed Data IP/MSC Datagroups/MSC SPM with Streamed Data over IP/MSC Datagroups/MSC EPM .................. 174
4.4.7 Comparison of Streamed Data over MPEG-2 TS/MSC Stream Mode with Streamed Data over IP/MSC Datagroups/MSC EPM ................................................................. 181
4.4.8 Comparison of Streamed Data IP/MSC Datagroups/MSC EPM with Streamed Data over IP/MPE-FEC .................................................. 183
4.4.9 Cached data over IP over MSC Datagroups over MSC SPM 185
4.4.10 Cached data over IP over MSC Datagroups over MSC EPM 188
4.4.11 Cached Data over IP over MPE-FEC ........................................ 190
4.4.12 Comparison of Cached Data IP/MSC Datagroups/MSC SPM with Cached Data over IP/MSC Datagroups/MSC EPM ........................................ 192
4.4.13 Comparison of Cached Data IP/MSC Datagroups/MSC EPM with Cached Data over IP/MPE-FEC .................................................. 198

4.5 Summary ................................................................. 198

5 Conclusions ............................................................... 201
5.1 The Contributions Of This Thesis ........................................ 201
5.1.1 Channel Performance ................................................ 201
5.1.2 Mathematical Modelling .............................................. 202
5.1.3 Software Simulation .................................................. 202
5.1.4 About the standards .................................................. 202
5.1.5 On the Impact of Errors ............................................. 204
5.2 Further Work .............................................................. 206

Bibliography .................................................................. 208

A Additional Results .......................................................... 215
A.1 Performance Metric .................................................. 215
A.1.1 Performance of a reliable channel, including transmission of overhead ................................................................. 215
A.1.2 Performance of a lossy channel with reliable reception after two transmissions, including transmission of overhead .... 216
A.1.3 Performance of a lossy channel after two transmissions, including transmission of overhead ................................................................. 216
A.1.4 Performance of a lossy channel needing increasing transmissions, including transmission of overhead ................................. 217
A.2 Mathematical Model .................................................. 218
A.2.1 DAB using SPM ....................................................... 218
A.2.2 DAB using EPM ....................................................... 220
A.2.3 IP DAB using SPM .................................................... 220
A.2.4 IP DAB using EPM .................................................... 220
A.3 Simulation Results .................................................. 220

B Design for Software Simulator ...................................... 246
  B.1 Database Design .................................................... 246

List of Acronyms .......................................................... 247
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The two numbers are encoded into an LT data unit</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>The second number is retrieved from the known first and encoded LT data unit</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>The three numbers are encoded into an LT unit</td>
<td>20</td>
</tr>
<tr>
<td>2.4</td>
<td>The second number is retrieved from the known first, third and encoded LT unit</td>
<td>20</td>
</tr>
<tr>
<td>2.5</td>
<td>MSC SPM Packet Length</td>
<td>28</td>
</tr>
<tr>
<td>2.6</td>
<td>MSC SPM Packet First/Last Flags</td>
<td>28</td>
</tr>
<tr>
<td>2.7</td>
<td>MSC Datagroup Type</td>
<td>33</td>
</tr>
<tr>
<td>2.8</td>
<td>The Meaning of the PLI Field Within an MOT Object Header Extension</td>
<td>36</td>
</tr>
<tr>
<td>2.9</td>
<td>The Meaning of the EXT Field Within an MOT Object Header Extension</td>
<td>36</td>
</tr>
<tr>
<td>3.1</td>
<td>Protocols And Their Attributes</td>
<td>68</td>
</tr>
<tr>
<td>3.2</td>
<td>Sizes of frames in an MPEG video stream</td>
<td>77</td>
</tr>
<tr>
<td>3.3</td>
<td>Cambridge All Zero Channel Configuration</td>
<td>78</td>
</tr>
<tr>
<td>4.1</td>
<td>Possible variables for a datagroups over MSC SPM network stack</td>
<td>86</td>
</tr>
<tr>
<td>4.2</td>
<td>Probability of the correct arrival, the Time Taken and the Channel Performance for a range of PDU sizes sent over a 384 kb/s MSC SPM TDC, affected by a Uniform Random BER of 0.0001</td>
<td>88</td>
</tr>
<tr>
<td>4.3</td>
<td>Possible variables for a datagroups over MSC EPM network stack</td>
<td>91</td>
</tr>
<tr>
<td>4.4</td>
<td>Probability of the correct arrival, the Time Taken and the Channel Performance for a range of PDU sizes sent over a 384 kb/s MSC EPM TDC, affected by a Uniform Random BER of 0.0001</td>
<td>93</td>
</tr>
<tr>
<td>4.5</td>
<td>Possible variables for a datagroups over MSC SPM network stack</td>
<td>96</td>
</tr>
<tr>
<td>4.6</td>
<td>Optimum Settings for datagroups for a given SPM packet size, derived from the simple mathematical model of a datagroups over MSC SPM channel</td>
<td>101</td>
</tr>
<tr>
<td>4.7</td>
<td>Optimum Settings for DAB EPM, derived from the simple mathematical model</td>
<td>107</td>
</tr>
</tbody>
</table>
4.8 Possible variables for IP over MSC Datagroups over MSC SPM network stack .............................................. 114
4.9 Smallest Found Optimum Settings for IP over datagroups over MSC EPM, derived from the simple mathematical model ............. 116
4.10 Optimum Settings for RTP over UDP over IP over datagroups over MSC EPM, derived from the simple mathematical model ............. 118
4.11 Statistical analysis of the effect of IP on carousel transmissions .... 144
4.12 Statistical analysis of the effect of IP on carousel transmissions .... 144
4.13 Statistical differences between the Simple and Enhanced Models for one datagroup over MSC SPM over a reliable channel ............. 149
4.14 Statistical differences between the Simple and Enhanced Models for one datagroup over MSC SPM over a reliable channel ............. 152
4.15 Statistical differences between the Simple and Enhanced Models for one 1,000,000 byte object over datagroups over MSC SPM over a reliable channel ......................................................... 153
4.16 Statistical differences between the Simple and Enhanced Models for one datagroup over MSC SPM over a reliable channel ............. 155
4.17 Statistical analysis of the comparison of observed simulation results with expected enhanced model results for streamed data sent over MSC Datagroups/MSC SPM ....................................................... 158
4.18 Statistical analysis of the comparison of observed simulation results with expected enhanced model results for carousel data sent over MSC Datagroups/MSC SPM ....................................................... 159

A.1 Reliable Performance Results. See figure 4.1 .............................. 215
A.2 Reliable Performance Results after two transmissions. See figure 4.2 216
A.3 Lossy Performance Results after two transmissions. See figure 4.3 216
A.4 Lossy Performance Results after multiple transmissions. See figure 4.4 217
A.5 Best-case Overhead Quantity for 24 byte SPM packets ................. 218
A.6 Best-case Overhead Quantity for 48 byte SPM packets ................. 218
A.7 Best-case Overhead Quantity for 72 byte SPM packets ................. 219
A.8 Best-case Overhead Quantity for 96 byte SPM packets ................. 219
A.9 Best-case Overhead Quantity for 24 byte SPM packets ................. 220
A.10 Best-case Overhead Quantity for 48 byte SPM packets ................. 220
A.11 Best-case Overhead Quantity for 72 byte SPM packets ................. 221
A.12 Best-case Overhead Quantity for 96 byte SPM packets ................. 221
A.13 Best-case Overhead Quantity for 24 byte SPM packets ................. 221
A.14 Best-case Overhead Quantity for 48 byte SPM packets ................. 222
A.15 Best-case Overhead Quantity for 72 byte SPM packets ................. 222
A.16 Best-case Overhead Quantity for 96 byte SPM packets ........ 222
A.17 Best-case Overhead Quantity for 24 byte SPM packets ....... 223
A.18 Best-case Overhead Quantity for 48 byte SPM packets ....... 223
A.19 Best-case Overhead Quantity for 72 byte SPM packets ....... 223
A.20 Best-case Overhead Quantity for 96 byte SPM packets ....... 224
A.21 Select results for streamed object over IP/MSC Datagroups/MSC
SPM/Cambridge All Zeros 1. See figure 4.73a ................. 224
A.22 Select results for streamed object over IP/MSC Datagroups/MSC
SPM/Cambridge All Zeros 2 part 1. See figure 4.73b ........... 224
A.23 Select results for streamed object over IP/MSC Datagroups/MSC
SPM/Cambridge All Zeros 2 part 2. See figure 4.73c ........... 225
A.24 Select results for streamed object over IP/MSC Datagroups/MSC
SPM/Cambridge All Zeros 3 part 1. See figure 4.73d ........... 225
A.25 Select results for streamed object over IP/MSC Datagroups/MSC
SPM/Cambridge All Zeros 3 part 2. See figure 4.73e ........... 226
A.26 Select results for streamed object over IP/MSC Datagroups/MSC
SPM/Cambridge All Zeros 4. See figure 4.73f ................. 227
A.27 Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge
All Zeros 1. See figure 4.76a .................................. 227
A.28 Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge
All Zeros 2 part 1. See figure 4.76b ......................... 228
A.29 Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge
All Zeros 2 part 2. See figure 4.76c ......................... 228
A.30 Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge
All Zeros 3 part 1. See figure 4.76d ......................... 229
A.31 Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge
All Zeros 3 part 2. See figure 4.76e ......................... 230
A.32 Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge
All Zeros 4. See figure 4.76f ............................... 230
A.33 Select results for 100000 Object over MPEG2-TS/MSC Stream
Mode/Cambridge All Zeros 1. See figure 4.79a ............... 231
A.34 Select results for 100000 Object over MPEG2-TS/MSC Stream
Mode/Cambridge All Zeros 2 part 1. See figure 4.79b ........ 231
A.35 Select results for 100000 Object over MPEG2-TS/MSC Stream
Mode/Cambridge All Zeros 2 part 2. See figure 4.79c ........ 231
A.36 Select results for 100000 Object over MPEG2-TS/MSC Stream
Mode/Cambridge All Zeros 3 part 1. See figure 4.79d ........ 232
A.37 Select results for 100000 Object over MPEG2-TS/MSC Stream
Mode/Cambridge All Zeros 3 part 2. See figure 4.79e ........ 232
A.38 Select results for 1000000 Object over MPEG2-TS/MSC Stream
   Mode/Cambridge All Zeros 4. See figure 4.79f ................. 233
A.39 Select results for streamed object over IP/MSC-FEC/Cambridge
   All Zeros 1. See figure 4.81a .................................. 233
A.40 Select results for streamed object over IP/MSC-FEC/Cambridge
   All Zeros 2 part 1. See figure 4.81b ............................ 233
A.41 Select results for streamed object over IP/MSC-FEC/Cambridge
   All Zeros 2 part 2. See figure 4.81c ............................ 234
A.42 Select results for streamed object over IP/MSC-FEC/Cambridge
   All Zeros 3 part 1. See figure 4.81d ............................ 234
A.43 Select results for streamed object over IP/MSC-FEC/Cambridge
   All Zeros 3 part 2. See figure 4.81e ............................ 235
A.44 Select results for streamed object over IP/MSC-FEC/Cambridge
   All Zeros 4. See figure 4.81f .................................. 235
A.45 Select results for carousel object over IP/MSC Datagroups/MSC
   SPM/Cambridge All Zeros 1. See figure 4.91a .................. 236
A.46 Select results for carousel object over IP/MSC Datagroups/MSC
   SPM/Cambridge All Zeros 2 part 1. See figure 4.91b ........ 237
A.47 Select results for carousel object over IP/MSC Datagroups/MSC
   SPM/Cambridge All Zeros 2 part 2. See figure 4.91c ........ 237
A.48 Select results for carousel object over IP/MSC Datagroups/MSC
   SPM/Cambridge All Zeros 3 part 1. See figure 4.91d ........ 238
A.49 Select results for carousel object over IP/MSC Datagroups/MSC
   SPM/Cambridge All Zeros 3 part 2. See figure 4.91e ........ 238
A.50 Select results for carousel object over IP/MSC Datagroups/MSC
   SPM/Cambridge All Zeros 4. See figure 4.91f ................. 239
A.51 Select results for carousel object over IP/MSC Datagroups/MSC
   EPM/Cambridge All Zeros 1. See figure 4.93a .................. 240
A.52 Select results for carousel object over IP/MSC Datagroups/MSC
   EPM/Cambridge All Zeros 2 part 1. See figure 4.93b ........ 240
A.53 Select results for carousel object over IP/MSC Datagroups/MSC
   EPM/Cambridge All Zeros 2 part 2. See figure 4.93c ........ 241
A.54 Select results for carousel object over IP/MSC Datagroups/MSC
   EPM/Cambridge All Zeros 3 part 1. See figure 4.93d ........ 241
A.55 Select results for carousel object over IP/MSC Datagroups/MSC
   EPM/Cambridge All Zeros 3 part 2. See figure 4.93e ........ 242
A.56 Select results for carousel object over IP/MSC Datagroups/MSC
   EPM/Cambridge All Zeros 4. See figure 4.93f ................. 243
A.57 Select results for carousel object over IP/MSC-FEC/Cambridge All
  Zeros 2 part 1. See figure 4.95a .......................... 243
A.58 Select results for carousel object over IP/MSC-FEC/Cambridge All
  Zeros 2 part 2. See figure 4.95b .......................... 244
A.59 Select results for carousel object over IP/MSC-FEC/Cambridge All
  Zeros 3 part 1. See figure 4.95c .......................... 244
A.60 Select results for carousel object over IP/MSC-FEC/Cambridge All
  Zeros 3 part 2. See figure 4.95d .......................... 245
A.61 Select results for carousel object over IP/MSC-FEC/Cambridge All
  Zeros 4. See figure 4.95e ................................. 245
## List of Figures

2.1 Conceptual diagram of the Forney Approach Convolutional Interleaver .................................................. 22
2.2 SPM Protocol Stack ........................................................................................................................................... 26
2.3 MSC SPM Packet Structure .............................................................................................................................. 26
2.4 EPM Protocol Stack ........................................................................................................................................... 29
2.5 FEC Frame Structure ........................................................................................................................................... 29
2.6 Bytewise Transmission Order for a FEC Frame ............................................................................................... 30
2.7 RS Packet Structure .......................................................................................................................................... 30
2.8 MSC Datagroup Header Structure .................................................................................................................... 31
2.9 The Transmission of MOT Segments .................................................................................................................. 32
2.10 MOT Object Header Core .................................................................................................................................. 34
2.11 MOT Object Header Extension .......................................................................................................................... 35
2.12 MOT Session Header Structure ......................................................................................................................... 37
2.13 MOT Session Segment Header Structure ........................................................................................................ 37
2.14 The Segmentation of MOT Objects ................................................................................................................ 38
2.15 MSC Datagroups over either MSC SPM or MSC EPM Protocol Stack ............................................................... 39
2.16 Encapsulation of an IP datagram in an MSC Datagroup .................................................................................. 40
2.17 Encapsulation of fragmented IP datagrams in MSC Datagroups ....................................................................... 40
2.18 Network Stack of Enhanced Stream Mode ....................................................................................................... 41
2.19 MPEG2-TS MUX Packet ................................................................................................................................ 42
2.20 MPEG-2 TS Packets with RS(204,188) Error Protection .................................................................................. 42
2.21 MPE Packet Structure ...................................................................................................................................... 43
2.22 MPE FEC Frame Structure ............................................................................................................................... 44
2.23 MPE FEC Packet Structure ................................................................................................................................ 44
2.24 Protocols used in DVB-h Stream Delivery ........................................................................................................ 45
2.25 Protocols used in DVB-h Cached File Delivery ................................................................................................ 46
3.1 The Simulation System Structure ....................................................................................................................... 76
3.2 The Route of capture of the all-zeros data sets ................................................................................................. 79
4.1 Reliable Channel including transmission of overhead ......................................................................................... 82
4.2 Reliable Performance after two transmissions, including transmission of overhead ........................................ 83
4.3 Lossy Performance after two transmissions, including transmission of overhead ........................................ 84
4.4 Lossy Performance after multiple transmissions giving reliable reception, including transmission of overhead ........................................ 85
4.5 The Effect of Uniform Random BERs on the probability of MSC SPM packets carried in an MSC SPM channel ........................................ 86
4.6 The Probability of Correct Receipt and the Time Taken for a range of PDU sizes over a 384 kb/s MSC SPM TDC ........................................ 87
4.7 The Channel Performance of a range of PDU sizes sent over a 384 kb/s MSC SPM TDC ........................................ 89
4.8 The Probability of Correct Receipt and the Time Taken for a 1,000,000 byte object transmitted over a range of PDU sizes over a 384 kb/s MSC SPM TDC ........................................ 90
4.9 The Effect of Uniform Random BERs on the probability of MSC SPM packets carried in an MSC EPM channel ........................................ 92
4.10 The Probability of Correct Receipt and the Time Taken for a range of PDU sizes over a 384 kb/s MSC EPM TDC ........................................ 92
4.11 The Channel Performance of a range of PDU sizes sent over a 384 kb/s MSC EPM TDC ........................................ 93
4.12 The Probability of Correct Receipt and the Time Taken for a 1,000,000 byte object transmitted over a range of PDU sizes over a 384 kb/s MSC EPM TDC ........................................ 94
4.13 The Channel Performance of a 1,000,000 byte object transmitted over a range of PDU sizes sent over a 384 kb/s MSC EPM TDC ........................................ 95
4.14 Performance of the entire datagroup range ............................. 96
4.15 A closer view of the saw teeth ........................................ 97
4.16 Performance of the entire datagroup range, given a channel affected by a uniform random BER of 10\(^{-4}\) ........................................ 98
4.17 Performance of datagroup payload sizes of 10 – 500 bytes, given a 10\(^{-4}\) Uniform Random BER ........................................ 99
4.18 Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a reliable channel ........................................ 100
4.19 Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of 10\(^{-6}\) ........................................ 101
4.20 A close up of an carousel rotation artefact, caused by the rounding in this model ........................................... 102
4.21 Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-5}$ ........ 102
4.22 Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-4}$ ........ 103
4.23 Improved view of the effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-4}$ ........ 104
4.24 Performance of the entire datagroup range .......................... 105
4.25 Performance of the entire datagroup range, given a channel affected by a uniform random BER of $10^{-4}$ .......................... 106
4.26 Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over datagroups over MSC EPM over a reliable channel ........................................ 107
4.27 Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over datagroups over MSC EPM over a channel affected by a uniform random BER of $10^{-4}$ ........ 108
4.28 The Comparison of the Channel Performance of an MSC SPM TDC with an MSC EPM TDC, for a Range of PDU Sizes ........... 109
4.29 The Comparison of the Channel Performance of an MSC SPM TDC with an MSC EPM TDC, for a Range of PDU Sizes ........... 110
4.30 The Comparison of the Channel Performance of a single PDU sent over an MSC Datagroups/MSC SPM TDC with a single PDU sent over an MSC Datagroups/MSC EPM TDC, for a Range of PDU Sizes ........................................ 112
4.31 The Comparison of the Channel Performance of a 1,000,000 byte object sent over an MSC Datagroups/MSC SPM TDC with a 1,000,000 byte object sent over an MSC Datagroups/MSC EPM TDC, for a Range of PDU Sizes .......................... 113
4.32 Performance of the entire datagroup range .......................... 115
4.33 Performance of the entire datagroup range, given a $10^{-4}$ Uniform Random BER ........................................ 116
4.34 Performance of DAB EPM of a 1,000,000 byte object, given a Reliable Channel ........................................ 117
4.35 Performance of one 1,000,000 byte object over RTP over UDP over IP over MSC EPM, given a Reliable Channel .......... 118
4.36 Performance of the entire datagroup range ................. 119
4.37 Performance of the entire datagroup range, given a 10\(^{-4}\) Uniform Random BER ........................................... 120
4.38 The Effect of Uniform Random BERs on the probability of MPEG2-TS packets carried in an MSC Stream Mode channel affected by varying .................................................. 121
4.39 The performance of a range of PDU sizes sent over an MPEGtwoTS/MSC Stream Mode channel .......................... 122
4.40 The performance of a range of PDU sizes sent over an MPEGtwoTS/MSC Stream Mode channel affected by a Uniform Random BER of 10\(^{-4}\) ........................................... 123
4.41 The performance of a 1,000,000 byte object sent over a range of PDU sizes over an MPEGtwoTS/MSC Stream Mode channel ...... 123
4.42 The performance of a 1,000,000 byte object sent over a range of PDU sizes over an MPEG2-TS/MSC Stream Mode channel affected by a Uniform Random BER of 10\(^{-4}\) ........................................... 124
4.43 Performance of the entire packet range .......................... 125
4.44 Performance of the entire datagroup range, given a 10\(^{-4}\) Uniform Random BER .......................................... 126
4.45 The performance of a 1,000,000 byte object streamed over a range of PDU sizes over an IP/MPE-FEC/DVB-h affected by a Uniform Random BER of 10\(^{-4}\) ........................................... 127
4.46 Performance of one 1,000,000 byte object over RTP over UDP over IP over MPE-FEC DVB-h, given a reliable channel ............ 128
4.47 Comparison of the performance of a single PDU carried over IP/MSC Datagroups/MSC EPM with a single PDU carried over MPEG2-TS/MSC Stream Mode ........................................... 130
4.48 Comparison of the performance of IP/MSC Datagroups/MSC EPM with MPEG2-TS/MSC Stream Mode .......................... 131
4.49 Comparison of the performance of a single PDU sent over MSC Datagroups/MSC EPM with a single IP packet sent over MPEFEC/DVB-h ......................................................... 132
4.50 Comparison of the performance of a 1,000,000 byte object sent IP/MSC Datagroups/MSC EPM with a 1,000,000 byte object sent over IP/MPE-FEC/DVB-h ........................................... 133
4.51 Comparison of the performance of IP over datagroups over MSC EPM with IP over MPE-FEC over DVB-h .......................... 134
4.52 Performance of the entire range of valid sizes of IP packets ............... 135
4.53 Performance of the entire range of valid IP packets, given a channel
affected by a uniform random BER of $10^{-4}$ .......................... 136
4.54 Improved view of the performance of the entire range of valid IP
packets, given a channel affected by a uniform random BER of $10^{-4}$ 137
4.55 Performance of IP over datagroups over MSC SPM of a 1,000,000
byte object, given a Reliable Channel ...................................... 138
4.56 Performance of IP over datagroups over MSC SPM of a 1,000,000
byte object, given a $10^{-4}$ Uniform Random BER ...................... 138
4.57 Performance of IP over datagroups over MSC EPM of a 1,000,000
byte object, given a Reliable Channel ...................................... 139
4.58 Performance of IP over datagroups over MSC EPM of a 1,000,000
byte object, with a channel affected by a Uniform Random BER of
$10^{-4}$ .................................................................................. 140
4.59 Performance of one 1,000,000 byte object over IP over MPE-FEC
DVB-h, given a channel affected by a Uniform Random BER of $10^{-4}$ 141
4.60 Performance of one 1,000,000 byte object over IP over MPE-FEC
DVB-h, given a channel affected by a uniform random BER of $10^{-4}$ 142
4.61 Comparison of the performance of IP over datagroups over MSC
SPM with datagroups over MSC SPM ........................................ 143
4.62 Comparison of the performance of IP over datagroups over MSC
EPM and datagroups over MSC EPM ........................................ 145
4.63 Comparison of the performance of IP over datagroups over MSC
SPM and IP over datagroups over MSC EPM .............................. 147
4.64 Comparison of the performance of IP over datagroups over MSC
EPM with IP over MPE-FEC over DVB-h ................................. 148
4.65 Comparison of the Simple Model with the Enhanced Model for one
datagroup over MSC SPM over a reliable channel ...................... 150
4.66 Comparison of the Simple Model with the Enhanced Model for one
datagroup over MSC SPM over a channel affected by a Uniform
Random BER of $10^{-4}$ ....................................................... 151
4.67 Comparison of the Simple Model with the Enhanced Model for one
1,000,000 byte object over datagroup over MSC SPM over a reliable
channel .................................................................................. 153
4.68 Comparison of the Simple Model with the Enhanced Model for a
1,000,000 byte object sent over MSC datagroups over MSC SPM
over a channel affected by a Uniform Random BER of $10^{-4}$ .... 154
4.69 Comparison of observed simulation results with expected enhanced model results for streamed data sent over MSC Datagroups/MSC SPM ...................................... 158

4.70 Comparison of observed simulation results with expected enhanced model results for carousel data sent over MSC Datagroups/MSC SPM ...................................... 159

4.71 Channel performance of a single IP packet over MSC Datagroups over MSC SPM over a channel affected by a Uniform Random BER of $10^{-4}$ ........................................... 161

4.72 Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC SPM over a channel affected by a Uniform Random BER of $10^{-4}$ ......................... 162

4.73 Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC SPM over the Cambridge all-zero channels ........................................ 163

4.74 The performance of IP packets over MSC Datagroups over MSC EPM over the channels affected by the Cambridge reception data ......................... 166

4.75 Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over a channel affected by a Uniform Random BER of $10^{-4}$ ............................... 167

4.76 Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels ........................................ 168

4.77 The performance of MPEG2-TS over MSC Stream Mode over the channels affected by the Cambridge reception data ................................................................. 169

4.78 The performance of a 1,000,000 byte sent over MPEG2-TS over MSC Stream Mode over the channels affected by the Cambridge reception data ........................................ 170

4.79 Channel performance of a 1,000,000 byte object over MPEG-2 TS/MSC Stream Mode over the Cambridge all-zero channels ......................................................... 171

4.80 Performance of a 1,000,000 byte object over IP over MPE-FEC over DVB-h affected by a Uniform Random BER of $10^{-4}$ ......................................................... 172

4.81 Channel performance of a 1,000,000 byte object over IP packets over MPE-FEC over the Cambridge all-zero channels ............................................. 173

4.82 Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1 ........................................ 175
4.83 Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 1 ............... 176
4.84 Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 2 ............... 177
4.85 Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 1 ............... 178
4.86 Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 2 ............... 179
4.87 Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1 ............... 180
4.88 Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels ........................................ 181
4.89 Channel performance of a 1,000,000 byte streamed object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels ........................................ 183
4.90 Channel performance of a cached 1,000,000 byte object over IP packets over MSC Datagroups over MSC SPM over a channel affected by a Uniform Random BER of $10^{-4}$ ............... 185
4.91 Channel performance of a cached 1,000,000 byte object over IP packets over MSC Datagroups over MSC SPM over the Cambridge all-zero channels ........................................ 186
4.92 Channel performance of a cached 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over a channel affected by a Uniform Random BER of $10^{-4}$ ............... 188
4.93 Channel performance of a cached 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels ........................................ 189
4.94 Performance of a cached 1,000,000 byte object over IP over MPE-FEC over DVB-h affected by a Uniform Random BER of $10^{-4}$ ........................................ 190
4.95 Channel performance of a carousel 1,000,000 byte object over IP packets over MPE-FEC over the Cambridge all-zero channels ........................................ 191
4.96 Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1 ............... 192
4.97 Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 1 ............... 193
4.98 Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 2 ............... 194
4.99 Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 1 ............... 195
4.100 Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 2 ............... 196
4.101 Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1 ............... 197
4.102 Channel performance of a 1,000,000 byte carousel object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels ......................... 199

B.1 Entity Relationship diagram of the DRBE database ............... 246
Chapter 1

Introduction

Broadcast radio has existed since the early twentieth century, providing a variety of programmes including music and spoken word, all using analogue transmission mechanisms. In the early 1980s there was a move to develop digital broadcasting. Work on the Eureka-147 project, more commonly known as Digital Audio Broadcast (DAB), started in 1987. Commercial and public broadcast DAB was implemented across several European countries, and commercial DAB receivers came to the market in 1999. Other broadcast digital radio standards, including Digital Multimedia Broadcast (DMB) and Digital Video Broadcast - Hand-held (DVB-h), have been developed more recently than, but concurrently with, DAB. These other standards have been developed to address some of the shortcomings that DAB can present. There will be further commentary on this in later chapters.

There are a number of common properties to all digital broadcast radio mechanisms. They all:

- are simplex broadcast networks
- have regional agreements (Europe, North America, Far East) on physical bandwidth location
- have national coverage at least planned in many countries provided by broadcast companies, or state institutions
- have an expanding consumer base on a national scale
- provide only broadcast mechanisms natively

1.1 Digital Radio Data Services

With the advent of digital channels many additional services to music and spoken word have evolved. It is these services that are the primary interest of this research.
Broadcast digital radio has a good, flexible, inherent data broadcast mechanism, allowing consumers to receive the same data at the same time, relatively cheaply, with efficient use of bandwidth.

As with all simplex channels, there is no back channel. This means that all available bandwidth is used for transmission, and all receivers have access to the same bandwidth at the same time. With a national network, it is possible to create a digital broadcast network that reaches hundreds of thousands of network nodes.

The nature of simplex communication also means that a receiver cannot request a retransmission if a packet is received in error, or not received at all. For data that is streamed, the only possible action a receiver may make is to recover the service as soon as possible, and accept small service glitches or service outages. In the case of cached data, everything must be retransmitted repeatedly to form some kind of channel reliability.

Each of the standards configuration parameters are constrained by the definition of the standards. This research was part funded by a commercial broadcast company called Arqiva (formally known as NTL Broadcast), who were interested in better understanding of the standards that they are delivering across Britain. There were two pieces of knowledge that they were particularly interested in: how best to tune the standards; and the impact of real errors on the performance of data carried by the standards. Part of this research was to run an experiment to capture real errors on a transmitter at Cambridge, to then test different protocols and tunings to see what performs best. The results of the tuning and the Cambridge capture experiments are presented in chapters 4 and 5.

There are a number of applications that broadcast radio may be used. There are public service and commercial radio stations, which many people are already familiar with. Broadcast digital radio can also be set up for a small geographical area, allowing for specialist applications, for example a large building site could use a local broadcast digital radio service to provide site information over a large area. Similarly a large scale emergency situation could employ a broadcast digital radio network to quickly pass information between emergency services and workers. There are also other existing broadcast radio services, like maritime radio, that may benefit from having better data service provision.

1.2 Alternative Technologies

The purpose of data services over broadcast digital radio is to provide useful services to hand-held devices. There are a number of alternative mechanisms that can provide these kind of services.
1.2.1 Mobile Telephony

Digital mobile telephony is grouped into Second Generation (2G), Enhanced Second Generation (2.5G) and Third Generation (3G). For the context of this research, all of these are able to carry data services alongside the more obvious voice communication.

It is noted here that although First Generation (1G) is able to carry data, by employing a modulator/demodulator (modem), that this is not a native data service mechanism, so is not analysed here.

Mobile telephone networks:

- are circuit switched networks
- have international agreement on physical bandwidth locations
- have international coverage provided by Telcos
- have an existing widespread consumer uptake of technology
- have limited scalability, due to the resolution of a network cell
- do not support broadcast or multi-cast protocols natively

1.2.2 Wireless Networking for Hand-held Computers

There are a number of wireless networking standards. Wireless Fidelity (WiFi) and Worldwide Interoperability for Microwave Access (WiMAX) are currently well-known complementary standards - WiFi is designed for high speed short range applications (in the traditional Local Area Network (LAN) geographical area), WiMAX as a broadband town-wide area (aiming to be a Metropolitan Area Network (MAN) solution). These standards provide an Ethernet network on 2.4 Giga-Hertz (GHz) (WiFi) and 2–66 GHz (WiMAX).

Wireless networks:

- are packet switched networks
- have international agreement on physical bandwidth location
- have no national coverage planned. There is limited coverage typically provided by private companies
- have wide consumer uptake as many new computer devices are sold with WiFi built in
• have a limited scalability, WiFi has a range of about 100 meters outside, dependant on the number of subscribers on the network. WiMAX is likely to perform well up to 5 miles from the base station, but the number of subscribers to that base station will effect the performance of the network.

• theoretically support both broadcast and multi-cast natively, however these are dependant on the providers' infrastructure, and broadcast and/or multi-cast services may not be provided.

1.3 Broadcast Versus Unicast

The broadcast communication technique works by sending all entities on a network exactly the same data concurrently. The unicast communication technique works by sending the information to the relevant entity only.

The merit of a unicast communication mechanism are that the bandwidth of the network is kept to the minimum required for one conversation. The communication is kept between two end points. However, this means that two identical messages are sent out, by a server, onto the network if two identical requests are made.

Broadcast techniques work by sending out one copy of the data to everyone, regardless of whether they want it. This leads to reduced bandwidth for individual transactions, but the advantage is that if many people want the same information, then there is less repetition on the network, releasing bandwidth for other uses.

The advantage of using a native broadcast network, like DAB, DMB or DVB-h, over alternative wireless technologies, like the mobile telephony networks, WiFi or WiMAX, for broadcast data is that all of the available bandwidth can be used for broadcast services. Therefore there is no conflict between broadcast services and unicast services or the use of the back channel.

1.4 Modelling and Simulation

When analysing systems it is useful to have tools that mimic real systems. There are two main ways to accomplish this:

**Modelling** is a process by which a mathematical formula is constructed to represent the system. These tend to be run quickly, but restrict the settings of the experiment

**Simulation** is a process by which an algorithm is constructed to represent the system. These tend to be run slowly, be can be highly flexible in altering
In creating both models and simulators, assumptions are made about the real system to reduce the complexity, and aid understanding of the attributes of the system.

One of the advantages of simulators is that real data may be analysed. Arqiva, formally known as NTL Broadcast and who part sponsored this work, facilitated the capture of real world data though the provision of bandwidth in a DAB multiplex and lent specialist reception equipment, which provided the mechanism for real World error profiles to be captured, and these are analysed in this work.

1.5 Contributions of this work

This work provides new mechanisms to model and simulate Digital Broadcast Radio, which has provided knowledge of the best configuration for the data broadcast mechanisms within DAB, DMB and DVB-h. In addition the performance of Forward Error Correction (FEC) mechanisms and the impact of data retransmission is analysed, which has shown the affect on the performance of the channel. To provide this knowledge, a new network metric has been developed, dubbed channel performance, to provide a fair comparison of performance of different standards and across different bandwidths, taking into consideration both the efficiency of the transmission and the loss rate caused by an imperfect channel. This has shown that an increase in strength of a FEC mechanism does not necessarily mean an increase in performance over a real data channel.

1.6 Thesis Structure

This chapter has outlined fields of interests for this research. Chapter 2 provides an overview of the digital broadcast mechanisms DAB, DMB and DVB-h. In chapter 3 there is a description of a mathematical model of DAB, DMB and DVB-h, and a broadcast digital radio transmission simulator capable of simulating various configurations of broadcast data standards. Chapter 4 provides results from both the mathematical model and the simulator. Finally this thesis concludes in chapter 5.
Chapter 2

Background

In the introduction chapter, there is an overview of the current broadcast radio standards, both at a high level and also some of the mechanisms used to construct the standards; there is also a discussion of ongoing and completed work relevant to this research.

2.1 Broadcast Information Systems

The first computers were mainframe machines which required local access, or at best local telephone infrastructure for a dumb terminal. In 1966 the first networks were developed to connect two mainframes together. This grew into Advanced Research Projects Agency Network (ARPANET), the World's first Wide Area Network (WAN) [55].

During computer infancy, only unicast was considered to pass information from one network node to another. Conceptually, unicast data transmission entails a connection that has only three entities involved - each end of the connection, and the route; where the route is thought of as a wire, and is therefore unimportant. This is a one-to-one relationship and the feasibility of positive feedback from one node to the other is high. The networking protocol Transmission Control Protocol (TCP) (TCP is normally sent over an Internet Protocol (IP) link and so is normally written as TCP/IP) works in exactly this way - for all data that is transmitted, an acknowledgement must be made [53]. If either the data or the acknowledgement is lost, then the data is sent again until either the data and acknowledgement both arrive correctly, or the mechanism gives up.

We can relate this to a normal telephone conversation, it doesn't matter how much technology is between the two people talking, they are talking to each other and are affirming that they are understanding each other. However there are a lot of situations where unicast communication is not efficient, for example if we had
to use a unicast mechanism in a meeting it could translate to everyone speaking only to the chair, and the chair having to confirm to a number of people that their understanding was correct.

Outside the sphere of computers, then, non-unicast networks are widespread. For instance, since 1922 the British Broadcasting Corporation (BBC) has been transmitting a public radio service [8]; one source and many listeners. This is an example of a broadcast network; a one to many relationship. The difference between a broadcast network and a multi-cast network is subtle; in a broadcast scenario everyone is bombarded with the information. A multi-cast network works similarly, except that you choose to receive the data. In a switched computer network where it is possible to send data to a specific item of hardware, it makes better sense to use multi-cast techniques, in a non-switched network, the only choice is to broadcast data. The problems of optimisation are not dissimilar, however.

There are situations where multi-cast networks would be best for computer networks. If we take the real-life radio analogy further we can apply it to Internet radio stations. The BBC transmit its programmes over the Internet in addition to using radio waves. In a unicast-only environment it would be necessary for every single user to have an individual connection to the BBC radio server. If the connection is over TCP/IP then there would be a confirmation for every packet that is sent from the server from every client. This is completely inappropriate, especially when the possibility that there may be many thousands of concurrent connections is taken into consideration; so confirmation is not sensible. If we use User Datagram Protocol over Internet Protocol (UDP/IP) where confirmations are not part of the protocol [51], the server still has to say the same thing many times to each client. It makes sense to have a repeater closer to the client so that the original server only has to talk to the repeaters, just like in the analogue broadcast radio network.

The consequence of having this repeater is that there is no way for the client to give positive (or negative) feedback, as the connection to the source is simplex. This means that another way has to be found to make the link between the server and the client reliable, as there is no manner to confirm that the data was correctly received, which is the desirable outcome of transmitting data. This is especially true when the data is transmitted over a lossy channel like Radio.

2.2 Literature Review

This section presents the background to this research. The areas covered are computer networks with particular interest of wireless connectivity, the metrics to
measure them, and the mechanisms to try to negate the impact of errors on data in broadcast networks.

2.2.1 Overview of Existing Wireless Digital Communication Technologies

This section looks at the available physical mechanisms to provide digital network connectivity to a range of devices.

Wireless Networks

Here we will look at mechanisms that provide network connectivity used commonly in computers.

The Institute of Electrical and Electronics Engineers (IEEE) 802.11x standards commonly dubbed WiFi, provide a wireless LAN connectivity for nodes on a computer network. As the 802.11 standards are wireless, the node may be stationary, portable or mobile [31]. The 802.11 network is intended for a geographical area of about 100 metres maximum distance from the base-station, or between devices, in very favourable conditions.

802.16 WiMAX While the 802.11 standards excel at LAN networks, their use is limited to about 100 metres in favourable conditions. WiMAX was designed with a greater geographic region in mind, suitable for use in both MAN and WAN configurations, providing broadband-speed connectivity [60].

Bluetooth is a Personal Area Network (PAN) radio communication protocol designed with mobile phones, Personal Digital Assistant (PDA) devices, Personal Computer (PC) peripheral devices and other such technology, in mind. It can be used to link two devices together, for connections akin to serial-cable communication between the devices, phones transferring business cards for example, or it can be used to connect devices to more serious networks, with the potential to provide a link to the Internet [70].

Infrared Data Association (IrDA) is an infrared network physical mechanism. Before bandwidth licensing and cost enabled wireless networks (like Bluetooth and WiFi) to become a possibility, the Infrared Data Association (IrDA) protocol was developed to provide serial-cable like data transmission between two devices. The IrDA protocol is flexible, being able to carry data, such as business cards, between phones up to providing a link to a laptop computer for connection to a network or the Internet. It can only connect
two devices together, and is limited to line-of-sight over distances of up to a metre [33].

**Mobile Phone Networks**

Mobile phone networks are prevalent across the planet. Although they were originally developed to provide a more flexible voice communication network, they quickly became capable of conveying data.

Mobile phone technologies have developed over time. Retrospectively, the stages in development have been grouped into *generations*:

1G cellular mobile phone networks first existed in the early 1980s. In spite of the status of 1G, there were earlier radio telephone methods, however these were not cellular techniques. The only way to transfer data across a 1G network is to use an analogue modem, an expensive and slow solution for reliable data transfer.

2G mobile phone networks appeared during the early 1990s.

2G technologies include:

- Global System for Mobile Communications (GSM)
- CdmaOne
- Digital Advanced Mobile Phone System (DAMPS)

2.5G mobile phone networks offer improved data services over the 2G network model, using General Packet Radio Service (GPRS). The GPRS network is Internet-aware, that is there is no special method to connect cellular devices to the Internet.

3G is a coverall term for number of different technologies, all of which conform to the International Mobile Telecommunications-2000 (IMT-2000) standard, laid out by the International Telecommunication Union (ITU), which is the leading United Nations agency for communication technology [62]. The IMT-2000 provides a flexible standard allowing for terrestrial and/or satellite based technology to create fast mobile data communication networks, and was designed with existing technology in mind.

3G technologies include:

- Universal Mobile Telecommunication System (UMTS)
- CDMA2000
- Enhanced Data GSM Environment (EDGE)
**GSM** is the European 2G solution, although there is migration towards Global System for Mobile Communications (GSM) within America. It uses Time Division Multiple Access (TDMA) to allocate when cellular devices may use the network to transmit and receive data. It incorporates encryption, data networking at 9.6 kilobits per second (kbps) via digital modems, Short Messaging Service (SMS) for text messages and paging services, Call Forwarding, Caller Identification, call waiting and multi-party conferencing. All of this means that the GSM 2G solution is more ready for advanced data services than the 1G offerings. GSM is the most secure of the 2G architectures. GSM provides a channel with a bandwidth of 9.6 kbps [36, Chapter 6].

**CdmaOne** is an implementation of the TIA-EIA-95 standard, which is also known as IS-95 and TIA-EIA-95. CdmaOne is a 2G system used predominately in North America, Korea and Japan. CdmaOne is specified to be a circuit switched data network with a bandwidth of 14.4 kbps [9, 36].

**DAMPS** known as TDMA in the USA, is a 2G offering used in Latin America, the USA, New Zealand, parts of Russia and the Asian Pacific. Currently Digital Advanced Mobile Phone System (DAMPS) services are being discontinued in America, with a migration towards GSM technology [1].

**GPRS** 2.5G mobile phone networks offer improved data services over the 2G network model, using General Packet Radio Service (GPRS). The GPRS network is Internet-aware, that is there is no special method to connect cellular devices to the Internet. GPRS provides a theoretical maximum of 172.2 kbps by allowing data to compete with telephony for time in the TDMA mechanism. This maximum bandwidth is derived on the assumption that there is no error correction and that the network provider would allow one cellular user to take the entire bandwidth. This is likely never to happen [36, 63].

**UMTS** is an evolution of GSM, which employs Wideband Code Division Multiple Access (WCDMA) [11], which is one of the enhancements to the Code Division Multiple Access (CDMA) mechanisms, to improve the throughput of the data. As GSM is a favoured 2G mechanism across Europe, Universal Mobile Telecommunication System (UMTS) is strongly favoured by Europe. The data bandwidth available in the UMTS standard is variable dependant on service and application, and the range is between 2megabits per second (mbps) and 4.75kbps [36, 65].
CDMA2000  CDMA2000, based on the CDMA IS-95 mechanism, includes enhancements which will allow for a possible 3mbps data channel [36].

CDmaOne, CDMA2000, and WCDMA techniques share the CDMA root mechanism, and have evolved similarly. When an innovation has been applied to one, often there has been work to include it in the others [54].

EDGE  technology works by improving the usage of the existing GSM carrier signal. This is achieved by changing the underlying modulation technique from a Gaussian minimum-shift keying (GMSK) as used in GPRS systems to a 8-phase-shift-keying modulation. With this improvement, the Enhanced Data GSM Environment (EDGE) system can transmit data at a maximum rate of 384 kbps. This is constructed of eight 48 kbps time slots. By comparison, GPRS can transmit using eight 14 kbps and GSM using eight 9.6 kbps time slots [36].

Global Maritime Distress and Safety System (GMDSS)

Due to the international nature of shipping, many international agreements have been reached on the use of technology for distress and other ship-to-ship and ship-to-shore communication.

One of the internationally agreed standards is the Global Maritime Distress and Safety System (GMDSS), a collection of technologies working in conjunction to provide mariners with safety information, and the ability to ask for help, should they need it. Two of these systems use digital broadcasting mechanisms: Digital Selective Calling (DSC) and NAVigational TEXt (NAVTEX).

DSC is used to initiate radiotelephone communication for ship-to-ship, ship-to-shore and shore-to-ship transactions. Every DSC transceiver has a unique number, similar in concept to a Media Access Control (MAC) address in network hardware [32]. The transceiver listens on maritime Very High Frequency (VHF) channel 80 for a signal containing its number. When the receiver is alerted by its number, the transmitter and receiver negotiate for a clear channel to communicate on, allowing the human operator to communicate.

Should the mariner need to send a Mayday distress call, the DSC transceiver can transmit this call on behalf of the mariner, allowing the mariner to perform other tasks, which is likely in a Mayday event [48].

NAVTEX At sea, navigation information is based on charts. New navigation warnings and information is more readily usable if presented pictorially. The NAVTEX system receives radio transmitted information, including images,
and prints them out. The information is categorised to allow the skipper to
decide whether it is relevant. The information is cached, so that the receiver
will only alert once for each message [48].

Broadcast Radio Networks

Broadcast radio networks providing audio services have been provided for many
years. More recently has been the mechanism to convey data using broadcast ra­
dio. Here we look at the standards that allow data in broadcast radio mechanisms.

Radio Data System (RDS) and Radio Broadcast Data System (RBDS) are competing systems for placing narrow bandwidth data over an analogue broadcast radio station signal. Radio Data System (RDS) is supported by the European Broadcasting Union (EBU) and Radio Broadcast Data System (RBDS) by the National Radio Systems Committee (NRSC) of the USA. RDS and RBDS typically carry information such as: Programme Identification, programme service name, alternative frequencies list, traffic programme identification, traffic announcement signals and Enhanced other station in­formation [71].

DAB is a broadcast technology designed to provide the option of a large number of different stations, or services, to transmit in the same physical bandwidth. The value of DAB is that it has cheap implementation and wide flexibility.

DAB is a digital broadcast standard. This means that the audio services pro­vided over may be viewed as a digital broadcast stream, with audio encoded using a particular codec mechanism, normally MPEG2 audio layer, although there are alternatives, and DAB with AAC+ (DAB+) uses the High Efficiency - Advanced Audio Coding (AAC+) codec. As DAB is digital natively, there are also mechanisms to provide other data services.

DAB is currently unreliable for data transmission. In this work we address its inadequacies and propose methods to improve its reliability: DAB is a broadcast mechanism, so natively has no return path for data or feedback on reception quality. There has been some research to provide a return channel.

Digital Video Broadcast (DVB) is a wide-bandwidth solution for digital televi­sion and data transmission. Digital Video Broadcast (DVB) has become a widely used standard for the transmission of public broadcast information across satellite, cable and terrestrial broadcast radio mechanisms [64].

There are different problems presented with the different mediums that DVB is carried over, which lead to a number of different standards, all based on the
core DVB standard, but which address the specific problems presented by its own medium: Digital Video Broadcast - Terrestrial (DVB-t), for example, was developed to cope with the inevitable extra noise and likelihood of being within the reception coverage of two transmitters providing the same service. This is a problem not encountered over a cable or a satellite channel [7].

**DMB** was an answer to the demand for video streaming over a channel that has a lower bandwidth than a DVB channel.

DMB uses the same underlying technology as DAB, so no new broadcast data technologies were developed. All of the data broadcasting techniques available to DAB may be employed in a DMB ensemble.

DMB emphasises use of the Enhanced Stream Mode transmission mechanism [19], which is used for video streaming in high loss environments [20].

**DAB+** utilises advances in audio codec mechanisms.

There have been advances in audio codecs beyond the MPEG2 audio mechanisms used natively in DAB. AAC+ is one such codec, with the advantage that it requires less overhead in the transmission stream than MPEG2 audio. DAB+ was developed as an enhancement of the standard DAB mechanism to utilise the benefit of a higher channel throughput [24].

### 2.2.2 Work Proceeding and Completed in the area

Eureka project 147, which was started in 1987 and completed in 2001, was a European-funded project, subsidised by a collaboration between a number of organisations based in Germany, France, the Netherlands and the United Kingdom [58]. The project's aim was to provide a viable high quality commercial digital broadcast radio mechanism, called DAB. DAB is now standardised as ETSI standard EN 300 401 [15]. By 1995 the BBC had decided that DAB radio was the future of broadcast radio services [69].

DAB was considered to be too narrow bandwidth for use in digital broadcast television, so in 1993 Digital Video Broadcast (DVB) was developed to meet this demand. There are a number of DVB derivatives to solve particular problems presented by different mediums: Digital Video Broadcast - Satellite (DVB-s) was developed in 1993, followed in 1994 by Digital Video Broadcast - Cable (DVB-c). These two standards assume that the channel is fairly reliable. Digital Video Broadcast - Terrestrial (DVB-t) was developed to replace terrestrial analogue broadcast television, and assumes that the channel will be affected by a greater degree of noise and the possibility of a receiver needing to cope with a
signal received from two different transmitters. DVB-h was an enhancement of DVB-t, which addresses the consumer demand for high bandwidth broadcast data to a hand-held device [7, 17].

The audio section of DAB uses an Moving Picture Experts Group (MPEG) layer 2 audio mechanism [15, Section 7], which can carry either single channel mono, dual channel mono, joint stereo or stereo audio data. The performance of DAB audio channels has been well researched, and a number of different recommendations were made to add other audio encoding mechanisms into the DAB standard [24, 3]. There are two notable enhancements to the DAB standard: DMB and DAB with AAC+ (DAB+). DAB+ is based on the DAB mechanism, but includes a different audio codec. In addition to MPEG layer 2 audio used in DAB, DAB+ can also employ High Efficiency - Advanced Audio Coding (AAC+). According to World-DMB research, an AAC+ transmission uses about 30% of the bandwidth in an equivalent bandwidth MPEG layer 2 audio transmission.

The mechanism for providing television services over a DAB ensemble has resulted in the DMB standard [19]. The performance was reviewed in Korea, where it was deemed to perform more reliably than an analogue with a greater capacity for alternative channels [38], with a level of network delay jitter deemed expectable by the MPEG 2 standard [37].

Some work has been done on the possibility of multi-standard receivers [56], including the possibility of some receivers making use of a back-channel. In the typical broadcast radio model, the receiver would not have a return path for information.

With the flexibility of digital channels has come the demand for multimedia content to enhance the experience of the audio services. Some work has been completed on how picture content may enhance an existing audio service, providing a clearer understanding to the user of the station about the subject being described [10].

There are a number of protocols that can be used to convey multimedia data. Amongst them is H.264 video compression, which previous work has evaluated over an Real-time Transmission Protocol (RTP) over User Datagram Protocol (UDP) over IP over DVB-h [66]. Here the authors present the effect of bit-errors on the DVB-h channel, and recommend that data broadcast over DVB-h will require the Multi-protocol Encapsulation (MPE)-FEC protocol, introduced in to the DVB protocol family by DVB-h [16].

The performance of broadcast data sent over MPE-FEC in a DVB-h channel has been researched, finding that the optimal IP packet size to be between 1,024 and 2,048 bytes in length [67]. Here the authors use a mathematical model of a urban reception to find the best size of IP packet to have the combination of
throughput and probability of reception.

Recently there has been research into the throughput of Multimedia Object Transfer (MOT) data in a terrestrial DMB channel [5]. This research attempted to discover the most optimum configuration, or tuning, for an MOT transmission by concentrating on the probability of receipt of the data, and concluded that the best mechanism was to use repetition to gain a relatively higher likelihood of correct data receipt. There has also been research into the performance of MPEG video data over wireless networks [39], showing that the burst error profile found in real wireless networks has a severe affect on the performance of such applications, with the need for correct tuning of the standard and finding best practice for use of FEC techniques.

### 2.2.3 Performance Metrics for Digital Broadcast Networks

Traditionally performance on a network is related to underlying loss and delay [36]. However, in broadcast networks the delay is less relevant as it is out of the control of the operator.

There is a significant difference between cached and streamed services over broadcast data services:

**Streamed data** is typically used to convey audio and visual data in a real-time manner. Many of the codecs that are used for this purpose are tolerant of errant data to some degree. Significant losses cause the data streams to fail, which the user is likely to find undesirable.

**Cached data** can be used to convey a variety of data types, all of which are unlikely to be used in a strict real-time manner. It must be assumed that the cached objects are intolerant of errant data, and therefore a lost Protocol Data Unit (PDU) will result in a lost object. Depending on what object is lost, it could mean that all of the data is unusable.

As the data is cached, it is possible for the data to be retransmitted over time. As there are normally many users of broadcast data services, it is unreasonable to expect retransmissions to be triggered by requests from receivers, regardless of whether any back-channel is available. The mechanism normally employed is for additional cycles of the data carousel to be transmitted, after an entire carousel transmission has been completed. The user perception of a loss in such a cached data mechanism is likely to be that the channel has a low bandwidth, and therefore undesirable.

From this we can take that any data arriving in error is undesirable, especially when the application employs a cached data mechanism.
2.2.4 Error Handling Techniques

In this section an error means a symbol, normally a character or a byte, that is received differently from how it was transmitted. The first task to achieve on reception is to discover whether any errors were within the data. Once this has been accomplished, we can look at the errant data and try to correct errors. There are several different classes of these error codes, and it would be useful to understand these different classes.

**Error Detecting Codes** or *check-sums*, are there to determine whether the data has arrived correctly or not.

Possibly the oldest computer implementation of this was the single bit parity check, where the algorithm was simply to count the number of bits that were transmitted as 1 (rather than 0). The final bit was set to ensure that there were an even or odd number of bits in state 1 (which was defined before transmission). This is not a particularly strong error detection code as it only takes an even number of bits in any message to be in error for the algorithm to break. There are alternative codes, which will be analysed later in this document.

**Erasure Correcting Codes** work by replacing missing data symbols, or collection of symbols. For these to work, any symbol that is received that is in error must be determined that it is indeed in error and deleted. In these algorithms, a symbol is normally an entity that is much bigger than a bit.

**Error Correcting Codes**, also known as FEC, work by discovering where the errored symbol or symbols are and providing enough extra data to determine what the error should have been. These only work if the entire stream is received, and the errors can be found within it. These codes have a finite tolerance; that is, they cannot correct the stream if there are too many errored symbols.

**Fountain Codes** are an extension concept over Erasure Correcting Codes. The definition of fountain codes is that they can recover the original symbols *(n)* from any set of the output symbols *(k)* with a high probability of success. *k* is always bigger than *n*; the performance of fountain codes is measured by how close *k* is to *n* [59, Section 1].

**Descriptions of Coding Mechanisms** There are a number of different aspects of error detection and correction technologies:
• Whether the original data is transmitted, or an encoded version. This is referred to as \textit{systematic} or \textit{non-systematic} codes, and is distinct from encryption.

• Whether the data is sent in fixed units, or a free flow form, known as either \textit{fixed rate} or \textit{rate-less} codes.

• Whether the data is sent in order or not, known as \textit{linear encapsulation} or \textit{interleaving} of the data, respectively.

**Systematic codes** are algorithms that work by adding correction data to the original data, that is that they transmit the data, as its original symbols, and some extra information. The extra data is there to allow the resurrection of the correct data symbols from the errant symbols.

**Non-systematic codes** are algorithms that work on top of the original data. The original data, as it's original symbols, are never transmitted. The correction data symbols and the original data symbols are merged.

**Fixed rate** algorithms are the traditional approach. There is a fixed number of data symbols that the algorithm will work on. If there is more data than is allowed by the fixed size, then additional frames will be sent, if there are two few, then the frame is padded prior to processing.

The rate would be \( \frac{k}{n} \) where \( k \) is the length (number of symbols) of the payload data and \( n \) is the length of the data that is to be transmitted.

**Rateless** algorithms do not have the restriction of having to fit within a frame, which means that they are a lot more flexible. Rather than structuring the payload, it requires a certain number of any transmitted symbols be received.

**Interleaving** is the process by which data is not sent sequentially. The symbols within the data stream are not sent in the order that they originally were in.

Although interleaving is not directly an error code, their value is in conjunction with error codes.

On their own, interleavers do not add reliability to a system, they only alter the order of the symbols, so if errors affect the channel, then the same amount of symbols that arrive in error, so the receipt reliability is the same. What interleaving does provide is a more useful error distribution.

A lot of real-life channels have the characteristic of errors occurring close together. This is called a burst error profile. Where the errors near each other,
the probability of an error correcting code fixing this problem is reduced as
many symbols will be missing.

Interleaving means that the errors will be spread throughout the stream, but
fewer errant symbols will be together. This allows the error correction codes
to function efficiently as it lowers the probability of the section of symbols
containing so many errors that the section cannot be corrected.

Some Implementations of Error Detecting and Correcting Codes

Here we look at the following actual codes that are relevant to this research; what
the existing codes are and how they are implemented.

- Cyclic Redundancy Check (CRC) codes
- Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes
- LT codes
- Online codes
- Viterbi Algorithm

CRC - Cyclic Redundancy Check. (CRC) s are based on polynomial arithmetic
mod 2, which is essentially the same as binary arithmetic mod 2, which is also
known as binary arithmetic without carrying [68].

In integer division, we have four entities: divisor, dividend, quotient and re­
mainder. In CRC terminology the divisor is called the generator polynomial (nor­
mally referred to as the polynomial); the dividend is the message - the wanted
data symbols to be sent; the remainder is the CRC and the quotient is waste.

The choice of the generator polynomial is quite important. In theory any
polynomial will work. In practice, some are much more reliable than others [61].

There are a number of known good CRC polynomials. DAB uses 16 bit CRC
fields in both its packet and datagroup entities. The two well-known 16 bit CRC
polynomials are:

- $x^{16} + x^{15} + x^2 + x^0$
- $x^{16} + x^{12} + x^5 + x^0$

Although creating a CRC is a repeatable process, it is not trivially reversible,
which places CRCs firmly in the error detection class of codes.

For a CRC to be of value it must be computed when the data is known to be
correct, and be available to the receiver to compare with the data. This means that
the CRC must be calculated before transmission, and transmitted with the data. When the data is received, the receiver performs the CRC process on the received data and compares the transmitted CRC with the one it has just generated. If the CRCs match, there is a high level of certainty that the received data is the same as the transmitted data; that there were no errors in the data.

In the DAB standard, CRCs are used to verify whether a packet has arrived in error, or not. They are also optionally used in the datagroup to verify that the datagroup has arrived correctly, although this is not normally implemented in practice.

**BCH and RS Codes** are fixed-rate error correcting codes.

Reed-Solomon (RS) codes are a sub-class of Bose-Chaudhuri-Hocquenghem (BCH) codes. RS code work on non-binary symbols, and are related to CRC codes.

If $p$ is a prime number and $q$ is any power of $p$, we have code symbols from a $q$-symbol alphabet. For any choice of positive integers $s$ and $t$, there exists a BCH code with length $n = q^s - 1$, which can correct any $t$ errors for $2st$ parity symbols. Therefore there are $n - 2st$ data symbols.

BCH codes can have any value of $s$. RS codes have the fixed $s$ value 1, which simplifies the equations somewhat:

$$
\begin{align*}
\text{Block length} & \quad n = q - 1 \\
\text{Number of data symbols} & \quad k = q - 1 - 2t \\
\text{Number of parity symbols} & \quad n - k = 2t \\
\text{Number of errors that can be corrected} & \quad t = \frac{n - k}{2}
\end{align*}
$$

The creation time for RS codes is $O(n^2)$. When being described, RS codes typically take the form RS $(n, k)$. There are a number of well known RS codes. In this research two are used: RS $(204,188)$ and RS $(255,191)$. RS $(204,188)$ is used within the DAB standard, and is a shortened version of the RS $(255,247)$ code. It is capable of correcting up to eight byte errors reliably. The RS $(255,191)$ standard is used within DVB-h and is capable of correcting up to 32 bytes in error reliably.

In addition to the normal RS code mechanism, there are two mechanisms that may be employed to adjust the efficiency of transmission and strength of error protection on the data: **puncturing** and **padding**. The padding mechanism works by reducing the amount of payload sent by the transmitter, with an agreement with the receiver that the receiver will add the data that was not transmitted, thus decreasing the efficiency of the transmission, whilst strengthening the error....
correcting potential of the RS code. Puncturing works by reducing the number of RS parity symbols transmitted, which has the effect of weakening the error correcting capabilities of the RS code, whilst increasing the channel efficiency [18].

LT codes were developed by Michael Luby [43] whilst working for Digital Fountain. They are a non-systematic, erasure-correcting code, based on the XOR function.

LT codes are protected by US patents [44] and [45].

If two Binary numbers are XORed together, and you know one number, you can XOR the known number and the XORed conglomerate, and you will discover the second number. Tables 2.1 and 2.2 demonstrate this principle.

Table 2.1: The two numbers are encoded into an LT data unit

<table>
<thead>
<tr>
<th>1 0 1 0 1 0 1 0</th>
<th>XOR</th>
<th>1 1 0 0 1 1 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 0 1 1 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: The second number is retrieved from the known first and encoded LT data unit

<table>
<thead>
<tr>
<th>1 0 1 0 1 0 1 0</th>
<th>XOR</th>
<th>0 1 1 0 0 1 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0 0 1 1 0 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This also works if there are more than one XORed file within the number, however there must be only one unknown number. This principle is demonstrated in tables 2.3 and 2.4.

Table 2.3: The three numbers are encoded into an LT unit

<table>
<thead>
<tr>
<th>1 0 1 0 1 0 1 0</th>
<th>XOR</th>
<th>0 0 1 0 1 0 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 0 1 1 1 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4: The second number is retrieved from the known first, third and encoded LT unit

<table>
<thead>
<tr>
<th>1 0 1 0 1 0 1 0</th>
<th>XOR</th>
<th>0 0 1 0 1 0 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0 0 1 1 0 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From this, it becomes apparent that if some of the numbers are known, then it can be possible to resurrect other numbers.

LT codes are sent in packets. The wanted data is split into pieces of a fixed size. Pieces of the wanted data are XORed with each other and identifiers of the comprising components of each section is sent with the data. This means that the receiver obtains the data in the form of an encoded block, and what was used to create that block. At this point it is certain that the data is not sent in its original state; therefore LT codes are non-systematic.

There are many different possible combinations of which blocks are used to create the encoded packet. This means that after a certain amount of time, it is then possible to recreate the original data. Luby states that after receiving typically 105% of the original data, then the data is recreate-able. This works because the data is continually being sent as a randomly chosen, different combination of blocks. This prevents us from stating the rate, it is not possible to exactly judge the quantity of reception by how much data is to be sent. Therefore this is an example of a rate-less algorithm.

It is clear that if there is an error within one of these packets, the entire mechanism will fail. Therefore LT codes need an error correcting or an error detecting code to work on the packets before the LT algorithm can be applied. Therefore LT codes must fall into the class of fountain codes, which are a subclass of erasure correcting codes.

The time taken to create the LT blocks is $O(\log n)$; the decoding time of an LT encoded stream is $O(n \log n)$, and the LT mechanism requires $n + O(\sqrt{n})$ blocks to be able to decode successfully [40].

**Online codes** are a refinement of LT codes. This means that Online codes are already defined as fountain codes, rate-less and erasure correcting [46].

The main difference between Online code and LT codes is the way that the component parts of the transmission blocks are chosen. Each transmission block is comprised of a number of component parts, which are chosen pseudo-randomly. The pseudo-random generator is a well-known, and consistent between the transmitter and the receiver.

Along with the data, some extra blocks are created; these blocks are called auxiliary blocks. The message blocks are reflected in three random auxiliary blocks, using the XOR process noted in tables 2.3 and 2.4. The number of auxiliary blocks will be 1% of the original message blocks.

Once these auxiliary blocks are created, the message and auxiliary blocks are combined, using the pseudo-random number generator to decide which of the blocks shall be combined together for transmission. In some cases this will be a
combination of only one block.

When these blocks are received, it is possible to reconstruct the message using an iterative process where the known is used to discover the unknown, again using the XOR technique.

As the algorithm is similar to LT codes, the metrics to compare are the encoding and decoding times, and number of blocks needed to reconstruct the original data. The creation time of Online codes is $O(1)$, and decoding time is $O(n)$, with a required $(1 + \epsilon)n$ number of blocks. That means that Online codes take a consistent time to encode, and a linear time to decode, whilst introducing low levels of overhead [46, Section 2].

The Viterbi algorithm is used to estimate state sequence information, providing the ability to predict the original sequence of bits in a data stream, to some level of accuracy [26].

Although there are a number of varying applications that the Viterbi algorithm has been applied to, within the field of broadcast digital radio it is used to protect the bits at the physical layer, to provide an error protected stream of bits. In the DAB standards, the base error protection is typically $10^{-4}$.

**Forney Approach to Convolutional Interleaving** is a well known technique to create a strong interleaving pattern to disperse error bursts more uniformly. This works by skewing the byte transmission order, with the receiver reversing the skew effect on receipt. Figure 2.1 shows the conceptual functionality of this interleaver.

![Figure 2.1: Conceptual diagram of the Forney Approach Convolution Interleaver][19, Figure 4]

Where:

$I$ The number of queues
The number of bytes to be sent in one Protocol Data Unit (PDU) is $N$.

The number of bytes in one unit of the queue, and is calculated as $\frac{N}{M}$.

The index of the interleaver queue is $j$.

The transmitter has $I$ rows. Each row has a different length queue: if the queue is referenced by $j$, then the queue length is $M \times j$ at the interleaver, and $M \times (I - 1 - j)$ at the deinterleaver.

The receiver also has $I$ rows, each with a different length queue. The main difference is that receiver has the opposite queue structure; the queue length can be found with the formula $M (I - j)$.

The impact of this is that a byte that has a small transmission delay has a long reception delay, and vice versa: either way, the total end-to-end delay is always $M \times I$.

The benefit that this provides is that there is a big separation between the bytes, which results in the intermingling of packets. Real errors tend to have a bursty error profile, and if the errors all arrive in one burst in one packet they may defeat the FEC protection. If the burst is spread across multiple packets, there is a much smaller chance that the FEC protection will be defeated.

### 2.2.5 Protocol Data Unit (PDU) Header Compression

In any channel it is necessary to add information to describe the data being carried. This information adds additional bytes, which is commonly called overhead. The more overhead there is, the worse the efficiency of the channel.

In modern wired networks, the amount of overhead introduced by the normal Transmission Control Protocol over Internet Protocol (TCP/IP) and UDP/IP headers is not considered so great that a reduction is necessary. In low bandwidth applications, especially those with a real-time application, the additional overhead may become too large a percentage of the data.

To fix this, there has been much work in creating compression for protocol headers. Here some of these developments are outlined.

The first steps to reduce the amount of overhead by removing superfluous information from Protocol Data Unit (PDU) headers carried over ARPA-Internet was started in 1984 [25]. This led, in 1990, to a proposed Internet standard to improve the performance of a low bandwidth TCP/IP channel, which is commonly known as Van Jacobson Compression (VJ Compression) [34]. At the time of writing, the author, Van Jacobson, did not believe that the added effort to carry UDP/IP was necessary, as his research indicated that UDP was not where the majority of the overhead was introduced. He found that the most significant
portion of the overhead was introduced in application protocols that employed UDP.

Following the development of RTP in 1996 [28, 57], it became clear that there was a need for header compression for Real-time Transmission Protocol over User Datagram Protocol over Internet Protocol (RTP/UDP/IP) channels. A standard was proposed in 1999, to provide better performance over low bandwidth links [4]. It was developed to cope with either a simplex or duplex channels, and over a simplex channels it was recognised that there would be a need to refresh the full header information periodically. Although it was not stated as such, it must have been assumed that the physical layer would be generally reliable. This assumption was derived by the tolerance for packet loss in this mechanism. If there were 16 packets lost, the decompressor might not notice that the channel was in error.

To address the reliability of a channel, a new standard was proposed in 2001, which was named ROBust Header Compression (ROHC) [2]. The first assumption that ROHC made was that it was likely to be employed carrying data over a wireless physical layer, and that wireless channels are generally considered to be lossy and error prone.

ROHC is outlined in more detail in section 2.3.8.

2.3 Understanding the Mechanics of the Standards

There is a range of different applications which require broadcast data channels, and these need different broadcast mechanisms to convey them. Some examples of applications that may be conveyed include: Broadcast Websites (BWS), streamed video and traffic information. This section looks at the mechanics of the protocols that convey generic applications. We will then go on to look at the performance of these protocols, which will provide an understanding of what effect these channels will have on the applications being carried by them.

The protocols of particular interest in this research are:

- Main Service Channel (MSC) Standard Packet Mode (SPM)
- MSC Enhanced Packet Mode (EPM)
- Datagroups over MSC SPM
- Datagroups over MSC EPM
- IP over Datagroups over MSC SPM
• IP over Datagroups over MSC EPM
• MPEG2 Transport Stream (MPEG2-TS) over MSC stream mode
• IP over MPE-FEC over DVB-h

2.3.1 DAB/DMB/DAB+ Ensemble

DAB, DMB and DAB+, which share the same underlying broadcast mechanism, have three data channels:

The synchronisation channel is used for: transmission frame synchronisation, automatic frequency control, channel state estimation, and transmitter identification. Although critical for the correct function of DAB and DMB, this channel carries no data that can be directly used by a user application, and so will be ignored for the rest of this research [15, Section 5.1].

The Fast Information Channel (FIC) is used to carry information that the receiver requires quickly. Amongst the information that is carried over the Fast Information Channel (FIC) is the Multiplex Configuration Information (MCI), which carries the data a receiver needs to correctly understand the components carried in the Main Service Channel (MSC). The purpose of this channel is primarily data that is generic across the stations being multiplexed together [15, Section 5.1].

The FIC also carries information that is critical to the correct functionality of a DAB or DMB transmission. However, the FIC does not carry data services, so will be ignored for the rest of this research.

The MSC is used to carry audio and data services. This is a time-interleaved channel, divided into a number of sub-channels, which in turn can be split into a number of services. The arrangement of these sub-channels and services is called the multiplex configuration [15, Section 5.1], and is provided by the MCI within the FIC.

The MSC, and the protocols which are carried by it are the subject of the following research.

The Main Service Channel (MSC) is transmitted within Common Interleaved Frames (CIF), which comprises of a number of Capacity Units (CU), all of which are 64 bits (8 bytes) long; therefore there are 864 CUs in a CIF. An integral number of the CUs make up a sub-channel [15, Section 5.3]. It is beyond the scope of this research to model this layer in the network stack, as there is no
possibility of altering this, so this is the last time that it will be mentioned in any depth.

Logical frames comprise of all of the CUs sent for a sub-channel in a 24ms burst, so a logical frame's size is dependant on the bandwidth that it is allocated.

### 2.3.2 MSC Standard Packet Mode (SPM)

There are two forms of packet mode: packet mode; and FEC protected packet mode. We will look at FEC protected packet mode in depth later, but it is worth noting at this stage that both use the mechanism outlined in this section. For clarity we will refer to them as Standard Packet Mode (SPM) for the unprotected method, and Enhanced Packet Mode (EPM) for the FEC protected method.

![Figure 2.2: SPM Protocol Stack](image)

MSC SPM packets are carried within logical frames, as can be seen in the network stack presented in figure 2.2. MSC SPM packets comprise of a packet header, body and footer. The packet data contains all or part of exactly one PDU from the protocols above it in the network stack. A diagram of an MSC SPM packet can be seen in figure 2.3.

![Figure 2.3: MSC SPM Packet Structure](image)

The mechanics of MSC SPM

MSC SPM packets may be 24, 48, 72 or 96 bytes in length, with the only restriction being the size of the bandwidth: the packets must fit integrally into the conveying logical frame, which size is affected by the bandwidth allocated to it. For MSC SPM, the bandwidth must be a multiple of 8kbps [15, Section 5.3.2].

Regardless of size, each packet contains 5 bytes of overhead, 3 being a packet header, and 2 being a 16 bit CRC carried in the packet footer.
Packet Length determines the length of the packet. The options for this field are listed in table 2.5.

Continuity Index is a modulus four counter to assert the sequence of packets within an address, as specified by the address parameter of the SPM packet header. This counter is always incremented by one for each packet transmitted.

First/Last states whether the receiver should expect more packets for this data-group. If datagroups are not being used, both bits are set to 0. The definition of how these flags are to be used is within Table 2.6.

Address is used to identify which service component, which are brought together into services [15, Section 6.1], is being transmitted in this packet. Address 0 is reserved and address 1111111110 is used for MSC EPM RS packets, leaving 1,022 addresses.

Command is used to indicate whether the packet contains part of the normal data being conveyed in this channel, or part of a special command, like Conditional Access (CA) data. When the command bit is set to 0, the data contained in the MSC SPM packet is considered to be part of the normal data being carried. When it is set to 1, the packet is assumed to be carrying special command data.

Useful Data Length defines, in bytes, the length of useful data within the packet. The remainder of the data space of the packet is padding. The maximum value for this field is a condition imposed by the Packet Length field, and the values are defined in the rightmost column of Table 2.5.

Useful Data Field contains the payload of the MSC SPM packet. The length of this field is specified in the Useful Data Length field.

Padding is the part of the Data Field that is not used by the Useful Data Length, where there are not enough bits to fill an entire Data Field, as MSC SPM packets are fixed sizes, as can be seen in table 2.5.

Packet CRC is the checksum for the MSC SPM packet. When analysed, if the packet does not have the same CRC as stated here, it will be dropped. This CRC is compulsory.
<table>
<thead>
<tr>
<th>Bit Combination</th>
<th>Total Packet Length</th>
<th>Packet Data Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_{15} )</td>
<td>( b_{14} )</td>
<td>(in Bytes)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>72</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>96</td>
</tr>
</tbody>
</table>

Table 2.5: MSC SPM Packet Length [15, Table 5]

<table>
<thead>
<tr>
<th>Bit Combination</th>
<th>The Packet is the:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_{11} )</td>
<td>( b_{10} )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.6: MCS SPM Packet First/Last Flags [15, Table 6]

The uses of MSC SPM

MSC SPM packets are used to convey many different mechanisms, which are covered later in this section. The protocols which may directly use MSC SPM are as follows:

**Transparent Data Channel (TDC)** is the mechanism by which any protocol may be conveyed using DAB. There are a number of different possible configurations that the Transparent Data Channel (TDC) may employ, and one of them is directly over MSC SPM [21, Section 4.1.1].

Other protocols may be carried over the TDC.

**Datagroups** may be carried over MSC SPM. This is covered in section 2.3.4. Other protocols are carried over MSC datagroups.

**2.3.3 MSC EPM**

MSC EPM is the short hand we are using to describe MSC SPM with FEC protection. It was introduced into EN300401 version 1.4.1 [15] to improve data services over DAB and DMB channels. It was designed to be backwards-compatible with MSC SPM in such a way that any receivers that cannot interpret the additional FEC data would still be able to interpret the MSC SPM, and correctly ignore the newly introduced packets which contain the FEC information.
The mechanics of MSC EPM

As our naming of the protocol suggests, MSC EPM mechanism is an enhancement on MSC SPM. All of the mechanisms employed by MSC SPM, which we have already seen in section 2.3.2, are also employed by MSC EPM. The MSC EPM network stack can be seen in figure 2.4.

![MSC EPM Stack Diagram](image)

**Figure 2.4: EPM Protocol Stack**

MSC EPM introduces a conceptual container, called a *FEC frame*, which is an array of 12 byte 204 bytes. The FEC frame is constructed of two parts: the *application data table*, which is 12 by 188 bytes; and the *RS data table*, which is 12 by 16 bytes. A FEC frame, with the application data table and the RS data table, can be seen in figure 2.5.

![FEC Frame Structure Diagram](image)

**Figure 2.5: FEC Frame Structure**

To construct the data correctly, the data to be carried by the MSC EPM mechanism is placed into the application table in columns. MSC SPM packets continue to fill the application data table until they have filled the application table integrally; an MSC SPM packet may never exist across two FEC frames, as this would break the MSC SPM compatibility.

Once the application data table has been filled, the FEC bytes are calculated across the rows of the FEC frame. This succeeds in a conceptual interleaving of the FEC data, improving the likelihood that the FEC information will successfully correct any error that it encounters. The FEC mechanism used is RS (204,188), which is a shortened version of RS (255,239) code [15, Section 5.3.5.1]. This code
is capable of correcting any eight bytes in error for 204 bytes transmitted, of which 188 bytes are payload. It is this choice of RS code that accounts for the width dimensions of the application data table and the RS data table.

Once the FEC frame has been established, it is transmitted, column by column, as shown in figure 2.6. The MSC SPM packets within the application table are transmitted over the logical frames with no further interpretation, as they already have their headers and footers. The RS data table has no header information, as they are all RS payload. To carry this, a RS packet is employed. The structure of an RS packet can be seen in figure 2.7.

The RS packet was designed guarantee to break the MSC SPM convention. This is safe development as the specification requires a receiver to quietly drop any packet that it does not recognise. The fields in an RS packet are as follows:

**Packet Length** is identical in location to the *Packet Length* field in an MSC SPM packet. An RS packet must be exactly 24 bytes long, and this field must be set to 00. This indicates to an MSC SPM receiver that it is a 24 byte packet, so will allow a receiver which is capable of only receiving MSC SPM data to gracefully ignore the correct number of bytes [15, Section 5.3.5.2].

**Counter** counts the number of RS packets sent for a given FEC frame. There is always nine RS packets sent, so this counter counts from 0 to 8.
Address this is set to 1111111110. This indicates that this is a RS packet, which will allow the MSC SPM only receiver to drop this packet [15, Section 5.3.5.2].

RS data field contains the RS bytes that were generated in the FEC frame construction.

Nine RS packets are transmitted per FEC frame, and each has a payload of 22 bytes. There are 192 bytes of RS payload to be transmitted in each RS data table, and there are 198 available bytes in the nine RS packets. This leaves 6 bytes in the last packet, which are left as padding bytes and ignored by the receiver.

The uses of MSC EPM

As MSC EPM is an enhancement of MSC SPM, the possibilities of utilisation, which were presented in section 2.3.2 are the same.

2.3.4 Datagroups over either MSC SPM or MSC EPM

MSC datagroups, which can also be referred to as MOT datagroups, are the native mechanism to DAB for conveying data. The structure of a datagroup can be seen in figure 2.8.

![Figure 2.8: MSC Datagroup Header Structure [15, Figure 9]](image-url)

Figure 2.8: MSC Datagroup Header Structure [15, Figure 9]
The mechanics of MSC Datagroups

Datagroups are carried over MSC SPM packets, which we reviewed in section 2.3.2. An MSC SPM packet may carry part or all of exactly one datagroup, which can be seen in figure 2.9.

![Figure 2.9: The Transmission of MOT Segments [12, Figure 18]](image)

The fields of MSC datagroups, as seen in figure 2.8, are as follows:

**Extension Flag** defines whether there is an Extension Field attached to this header. This bit is set to '1' if the extension field is present [15, Subsection 5.3.3.1].

**CRC Flag** defines whether there is a CRC checksum at the end of this Data Group. This bit is set to '1' if the CRC is present [15, Subsection 5.3.3.1].

**Segment Flag** defines whether there is a Segment Field, within the Session Header. This bit is set to '1' if the segment field is present [15, Subsection 5.3.3.1].

**User Access Flag** defines whether there is a User Access Field, within the Session Header. This bit is set to '1' if the user access field is present [15, Subsection 5.3.3.1].

**Data Group Type** is used to indicate the type of data carried within the Data Group Data Field. The meanings of some of the states of this section are given within Table 2.7.

**Continuity Index** is a modulus 4 counter that is incremented from the last time a datagroup with the same type was transmitted.

**Repetition Index** contains the number of iterations of this datagroup remaining on the carousel. If this field contains $111_2$, the implication is that this datagroup will be repeated infinitely, where infinity is any number over $15_{10}$. 
Extension Field is an optional field, its presence is dictated by the Extension Flag. This bit is set to ‘1’ if the extension field is present [15, Subsection 5.3.3.1].

Datagroups have headers incorporated into their structure, and an optional 2 byte footer, which contains the CRC for the entire datagroup header and datagroup data which consists of segment header and segment data.

<table>
<thead>
<tr>
<th>Bit state</th>
<th>The meaning is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>b₃ b₂ b₁ b₀</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>General Data</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>Conditional Access (CA) Messages</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>General Data and Conditional Access (CA) Messages</td>
</tr>
<tr>
<td>Other</td>
<td>Other purpose, defined within the DAB standard [15]</td>
</tr>
</tbody>
</table>

Table 2.7: MSC Datagroup Type [15, Section 5.3.3.1]

The Uses of MSC Datagroups

There are a number of standards that make use of MSC Datagroups. We saw that Transparent Data Channel (TDC) could be carried over MSC SPM or MSC EPM in section 2.3.2; TDC over either MSC SPM or MSC EPM is also a valid channel.

Both the MOT header mode and the MOT directory mode mechanisms use MSC Datagroups as the principle protocol to convey their data:

MOT Header Mode is used for broadcast data mechanisms where the object is used, and then discarded when the next object unaffected by error is received, or the object expires.

An example of this kind of mechanism is the MOT slide show, where content which is useful to a radio programme is transmitted alongside the audio [12].

MOT Directory Mode is used for broadcast data mechanisms where a set of objects are required for the user application.

An example of this kind of mechanism is the BWS model, where a number of files need to be available to the browser [12].

The Mechanics of MOT Header Mode over Datagroups

An MOT Header Mode transmission is made up of a number of data entities, which are referred to as Objects. These objects are arranged into a repeating entity, known as a carousel. The carousel may be repeated many times to increase
the likelihood of the data arriving at the receiver. Each repetition of a carousel is referred to as a rotation.

Objects carry the application level data within them, each having an MOT object header immediately preceding them. The MOT header comes in two parts, the mandatory part, called the header core, and an optional extension, called the header extension. The MOT Object structure can be seen in figures 2.10 and 2.11.

![MOT Object Structure Diagram](image)

**Figure 2.10: MOT Object Header Core [12, Figure 4]**

- **Body Size** defines the size of the Object to be sent. The information is described in this header core. (Objects can be files, streaming data, etc.) If the Body Size is set to all ones, it is assumed that the size of this object is unknown at the start of transmission.

- **Header Size** defines the total size of the header in bytes. This can range from 7 to 4,095.

- **Content Type** defines the general type of the data to be transmitted, whether it's text, audio, images, etc. A full list of these can be found in the DAB MOT standard [12, Table 1].

- **Sub Content Type** defines the specific type of the data to be transmitted. This says whether it's a HTML or plain text document - both subtypes of the Content Type Text as defined in **Content Type**. A full list of these can be found in the DAB MOT standard [12, Table 1].

- **PLI** is a two bit field to indicate the type that this parameter is. The Parameter Length Indicator (PLI) field may be set to one of four combinations, shown in Table 2.8.

- **Parameter Identifier** defines what this parameter is for. This is always 6 bits. A full list of the combinations can be found in the relevant standards [12, Table 2].
Figure 2.11: MOT Object Header Extension [12, Figures 5 & 6]
**EXT** is used only when the PLI field is set to $3_{10}$, i.e. "$1\ 1\ 1$". Its purpose is to indicate the length of the data field length indicator field. The meaning of the Extension Flag (EXT) bit is defined in table 2.9.

**Data Field Length Indicator** is used only when the PLI field is set to $3_{10}$, i.e. "$1\ 1\ 1$". It is used to indicate the length of the length if the Data Field. This field is either an unsigned 7 or 15 bit number, dependant on the EXT field.

**Data Field** is used only when the PLI field is set to $3_{10}$, i.e. "$1\ 1\ 1$". It contains the parameter data.

<table>
<thead>
<tr>
<th>Bit States</th>
<th>The total parameter size is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>one byte; no data field available</td>
</tr>
<tr>
<td>0 1</td>
<td>two bytes; of which the data field is one byte</td>
</tr>
<tr>
<td>1 0</td>
<td>five bytes; of which the data field is four bytes</td>
</tr>
<tr>
<td>1 1</td>
<td>defined in the Data Length Field. The maximum parameter length is 32,770 bytes</td>
</tr>
</tbody>
</table>

Table 2.8: The Meaning of the PLI Field Within an MOT Object Header Extension [12, Section 5.2.1]

<table>
<thead>
<tr>
<th>Bit State</th>
<th>The Data Field LengthIndicator is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7 bits long</td>
</tr>
<tr>
<td>1</td>
<td>15 bits long</td>
</tr>
</tbody>
</table>

Table 2.9: The Meaning of the EXT Field Within an MOT Object Header Extension [12, Section 5.2.1]

MOT mechanism uses the session information within the session header of the datagroup to provide version information for the Object. This is done so the receiver knows what data is contained within the datagroup data field. The structure of a session is shown in figure 2.12.

**Last Flag** defines whether this is the last segment, or that there are more segments to follow. If this is the last segment, this bit is set to 1 [15, Section 5.3.3.2].

**Segment Number** is a fifteen bit unsigned integer counter for segments. This counter is incremented by one at the start of each segment. This means that segments are transmitted with unique number within the carousel.

**Rfa** stands for *Reserved for future additions.*
Transport ID flag defines whether there is a Transport ID field included towards the end of this header. This bit is set to 1 if the transport ID field is present [15, Section 5.3.3.2].

Length Indicator is a 4 bit unsigned value that indicates the combined size, in bytes, of the Transport ID' field and the End User Access Field. The possible range for this field is between 0 and 15, although if the transport ID flag is present, this must be at least 2.

Transport ID is used to uniquely identify one object, with object header, entity. The entity consisting of object and object header is referred to as a data object.

End User Address Field is used to identify the address of the end user.

The Objects, and their headers, are split into segments, which can be seen in figure 2.13.
Repetition Count is a counter to indicate how many more times that the current object will be repeated. If this is set to seven, there are an infinite number of repetitions, where infinite is any number greater than six.

Segment Size holds the size, in bytes, of the Segment Data to be transmitted. This can range from 0 to 8,189.

Each segment may only encapsulate one entity, so the object header is kept distinct from the object body in this process. This can be seen in figure 2.14. Each MSC Datagroup may carry exactly one segment [12]. This means that for one MOT Header Mode object transmission, there is at least two datagroups sent: one for the Object header information, and one for the Object body. The header and the body will separately be split into as many segments as are necessary to be sent over the maximum available MSC Datagroup payload size. The specification states that the segments will be a uniform size, apart from the last one, which may be shorter: there is no facility for padding bytes in the MOT segment or MSC datagroup standards [12, 15].

Each segment received uninfluenced by error may be cached by the receiver. This means that if a segment is retransmitted many times, there is a greater chance of arrival. Both the MOT standard, and the MSC datagroup standards have the facility for repetition of their PDUs. At the datagroup level, the repetition means that the same datagroup will be transmitted a number of times after it has been transmitted. At the MOT level, the repetition means that group of datagroups
used to transmit the MOT object header and body will be repeated in the same pattern after the entire groups has finished being transmitted. Both can be used at the same time, and it is possible to assign different repetition levels to the different MSC datagroups.

The Mechanics of MOT Directory Mode over Datagroups

Much of the MOT Directory Mode mechanism is similar to the MOT Header Mode mechanism, which we have already reviewed in section 2.3.4. A significant difference is that MOT Directory Mode uses a directory object, containing all of the MOT Object headers, rather than transmitting the MOT Object headers next to the Object bodies [12].

This mechanism, therefore, is more suited for sustained data over periods of time, like a BWS.

2.3.5 IP over Datagroups over either MSC SPM or MSC EPM

IP is a prevalent network protocol, and it is well understood [52]. The wide uptake of IP means that there is a large variety of hardware solutions for IP based networks. The carriage of IP across any network is desirable for this reason [14].

The mechanics of IP over MSC Datagroups

The mechanics of IP are well understood [52], and we have already reviewed the transmission mechanisms used to convey datagroups over MSC SPM in section 2.3.2. This section concentrates on the manner in which an IP packet is transported over MSC datagroups. The network stack for IP over MSC Datagroups can be seen in figure 2.15.

Figure 2.15: MSC Datagroups over either MSC SPM or MSC EPM Protocol Stack

An MSC datagroup may carry all of exactly one IP packet. This is shown in figure 2.16.
The maximum size of an IP packet is 65,535 bytes [52], which is bigger than the maximum payload of MSC datagroups, which may have a maximum size of 8,189 bytes [15]. If an IP packet is ever larger than the MSC datagroup which will be employed to convey it, the IP packet will fragment into as many MSC datagroups as are required to transport all of the large IP packet. The IP packet will be fragmented in the well-understood manner [52]. If the datagroup payload size is specified to be less than the IP packet size, the IP packet will fragment, so although it is likely that the maximum size of an IP packet will be 1,500 bytes [30], the fragmentation mechanism may be necessary, as it’s conceivable that the broadcaster may choose a small MSC datagroup payload size.

Figure 2.17 shows the IP fragmentation process in context with MSC datagroups.
The uses of IP over MSC Datagroups

There are a number of uses for IP over MSC Datagroups. Any broadcast or multicast mechanism can easily be translated to a DAB data broadcast mechanism. Using IP within a normal network stack means that there is a wealth of tools, both software and hardware, which can be used within the transmission mechanisms and the receivers. Typical mechanisms might include UDP and RTP.

There has been some research done, known as the BT movio trials, which utilised an IP/MSC Datagroups/MSC EPM mechanism to transmit video data.

2.3.6 MPEG2 Transport Stream (MPEG2-TS) over MSC Stream Mode

DAB and DMB ensembles are capable of a native streamed data mechanism. This is called MSC Stream Mode, and is distinct from MSC SPM. Stream mode is carried in the MSC, within logical frames.

The mechanics of MPEG2 Transport Stream (MPEG2-TS) over MSC Stream Mode

MSC Stream Mode uses a mixture of interleaving and forward error correction to provide a resilient channel for streamed data transmission.

Figure 2.18 shows the mechanism by which MPEG2 Transport Stream (MPEG2-TS) packets are conveyed on a MSC. This mechanism was borrowed from the DVB standard [19, 16]. DVB is discussed in section 2.3.7.

![Network Stack of Enhanced Stream Mode](image)

The MPEG2-TS packets, referred to as transport MUltipleX (MUX) packets, are grouped into eights. With the exception of the synchronisation byte at the start of each MPEG2-TS packet, these are then pseudo-randomly adjusted to provide a better energy distribution. The pseudo-random sequence is re-seeded
every eight packets. The synchronisation byte of the MPEG2-TS packet, which is defined as $47_{\text{HEX}}$ is inverted for the first packet in the sequence of eight, to provide the receiver a mechanism to start the pseudo-random number sequence. Once this adjustment has been made, the RS (204,188) mechanism is applied to the MPEG2-TS packet. The structure of the RS error corrected packets can be seen in figure 2.20.

The RS error corrected packets are now regarded as a stream. They are then transmitted using a convolutional byte-wise interleaver, based on the Forney approach as previously seen in section 2.2.4.

In MSC Stream Mode, the convolutional interleaver has:

- $I = 12$ The number of queues
- $N = 204$ The number of bytes to be sent in one PDU
- $M = \frac{N}{I} = 17$ The number of bytes in one unit of the queue

**The uses of MPEG2-TS over MSC Stream Mode**

The official stream mode mechanism for all DAB based broadcast data standards is MPEG2-TS/MSC Stream Mode. This is used particularly in DMB to convey video transmissions, although it can conceivably be used for other streamed data broadcast transmissions as well.
2.3.7 IP over MPE-FEC over DVB-h

Development of DVB, a digital broadcast network capable of carrying digital television, started in 1993 [6]. There are a number of variants of DVB: DVB-c, DVB-s, DVB-t and DVB-h. This research focuses on DVB-h, as DVB-h was designed to be an alternative to DAB.

DVB-h is an extension to DVB-t to provide services better suited to hand-held devices, including mobile phones and Personal Digital Assistants (PDA). Among the additions that DVB-h provides are: the addition of a 4kbps bandwidth to the existing DVB-t 2kbps and 8kbps channels; and the MPE-FEC data broadcasting mechanism [17, 47]. There are a number of viable protocols carried over DVB-h. This research focuses on MPE-FEC, as this is equivalent to the DAB MSC SPM mechanism.

The mechanics of IP over MPE-FEC over DVB-h

DVB uses the International Organisation for Standardisation (ISO) MPE mechanism to carry data services. Typically DVB standards use IP over MPE. The MPE packet can be seen in figure 2.21. Only one whole IP packet, or one fragment as defined by the IP standard, may be carried in an MPE packet. The MPE packet stream is carried over MPEG2-TS packets.

![MPE Packet Structure](image)

Figure 2.21: MPE Packet Structure as defined by ISO/IEC 13818-6 [18, Section 7.1]

In the DVB-h standard the MPE mechanism uses the optional FEC mechanism, and as such is known as MPE-FEC. The DAB designers used this MPE-FEC mechanism as the origin design for the FEC frame within DAB MSC EPM, which we have seen in section 2.3.3.

The MPE-FEC frame, which can be seen in figure 2.22, comprises two tables: the application data table and the RS data table. Although the dimensions of the MPE-FEC frame are different, the way that the MPE-FEC frame is constructed is identical to that of the MSC EPM FEC frame, noting that contents of MPE packets, which in this case are IP packets, are used instead of MSC Datagroups.

MPE-FEC utilises a different RS code than that used in DAB MSC EPM, which account for the difference in widths of both the application data table and the RS data table. The RS code that is used in MPE-FEC is RS (255,191), which
means that for each 255 bytes transmitted, 191 bytes are potential payload and 32 byte errors are guaranteed to be correctable.

Once the MPE-FEC frame has been generated, it is transmitted in a similar manner to the MSC EPM FEC frame. The MPE-FEC frame application table is transmitted using MPEG2-TS packets, as seen in figure 2.19.

The RS data table is carried in MPE-FEC packets, which can be seen in figure 2.23.

There are two interesting features provided by the MPE-FEC protocol:

The padding columns field in the MPE-FEC packet header, shown in figure 2.23a, provides the mechanism to improve the FEC capabilities of the RS coding.

The FEC capabilities may be improved in MPE-FEC by the inclusion of padding columns to the application data table. These columns do not get transmitted; the MPE-FEC headers carry how many of the columns
of padding were included, the transmitter ignores these columns, the receiver reintroduces them and marks them as reliable. The RS mechanism can then correct up to 64 bytes in error, dependant on the level of padding chosen [18, Section 9.3.3.1]. The padding bytes are always assumed to be the last, the most right, columns in the application data table, and may be altered every frame.

The last section number field in the MPE-FEC packet header, shown in figure 2.23a, provides the mechanism to puncture the RS code, thus reducing the overhead of the MPE-FEC frame transmission. The punctured columns are always assumed to be the last, the most right, columns in the RS data table [18, Section 9.3.3.2].

RTP over UDP over IP over MPE-FEC

Although the MPE mechanism within the DVB channel is capable of conveying different protocols, here we will concentrate on the one in common usage within DVB-h: RTP/UDP/IP/MPE-FEC.

These protocols are familiar to duplex networking. The DVB-h stack can be seen in figure 2.24 [22].

![Figure 2.24: Protocols used in DVB-h Stream Delivery [23, Figure 18]](image)

The RTP, UDP and IP protocols all behave as they are defined in their standards [57, 51, 52, 22]. Each IP packet containing the RTP and UDP information will be placed into an MPE packet. The FEC mechanism, RS (255,191), is employed to create the FEC packets, allowing the receiver to correct up to $32r$ errors in the FEC frame, where $r$ is the number of conceptual MPE-FEC rows in the FEC frame.

The DVB-h MPE-FEC standard does not permit MPE sections to be longer than 4,096 bytes in length, which limits the entirety of the IP packet length to 4,083 bytes, as 13 bytes are the MPE packet header. This is longer than the standard IP on Ethernet Maximum Transmission Unit (MTU) of 1,500, but still only about half of the proposed jumbo frame length, which is loosely defined as 9,000 bytes in length.
Cached File Delivery

We saw in section 2.3.4 that the native mechanism to convey cached data within DAB is the MOT carousel. The common mechanism to convey data in DVB-h is File Delivery over Unidirectional Transport (FLUTE)/UDP/IPMPE-FEC [49, 41, 42, 22].

There are two ways that FLUTE may be used to convey carousels: static carousel mechanism and dynamic carousel mechanism.

The static carousel mechanism assumes that the same data will always be sent in future rotations of the carousel. This allows the receiver to sleep i.e. to stop receiving new data once the entire carousel has been received, until it expires, thus saving battery life of portable devices [22].

The dynamic carousel mechanism assumes that there may be minor revisions of the objects sent in a carousel in future rotations of the carousel. This allows the receiver to pick up alterations in objects, add or delete objects, whilst keeping the unchanged parts of the carousel, at the cost of the battery life of a portable receiver [22].

The FLUTE network stack is shown in figure 2.25, and is the same for both the dynamic and the static carousel mechanisms.

```
| FLUTE          |
| ALC/LCT       |
| UDP           |
| IP            |
| IPMPE-FEC     |
| DVB-h         |
```

Figure 2.25: Protocols used in DVB-h Cached File Delivery [23, Figure 18]

2.3.8 ROHC

The ROHC mechanism can carry data in either a simplex or duplex manner, and is explicitly designed to cope in an error-prone channel, or a channel susceptible to long a Round-Trip Time (RTT) where a acknowledgement based protocol like TCP may ordinarily fail due to a timeout [2, 35].

An ROHC compressor/decompressor can be in one of three possible states: *Initialisation and Refresh (IR)*, *First Order (FO)* and *Second Order (SO)*.
Initialisation and Refresh (IR) state is used to begin a conversation between compressor and decompressor. It sends all of the header information in uncompressed form, with additional ROHC information. The compressor will stay in Initialisation and Refresh (IR) state until it is fairly confident that the decompressor has received all of the data correctly.

IR state has no compression, and as it transmits additional information, is slightly worse than when ROHC is not employed. This state will be able to make the transition to either First Order (FO) or Second Order (SO) state, where the improvement in efficiency can be made.

First Order (FO) state is used to efficiently communicate the irregularities in the data stream.

The compressor/decompressor enter First Order (FO) state from either the IR state, when the compressor has adequate certainty that the decompressor can accurately decompress the stream, or from SO state, when there is an irregularity in the data being sent, which the decompressor will need to be informed of.

FO state has a better compression ratio than is found in IR state, but still carries a lot of additional information, making FO state less well compressed than SO state.

Second Order (SO) state is used when the data being sent on the channel is completely predictable, and the compressor is sufficiently confident in FO state that the decompressor will be able to correctly decompress the headers.

Second Order (SO) state has the most optimum compression rate, however an error in the received data in FO state will result in the loss of all packets at SO state, until the mechanism refreshes.

ROHC has three modes of operation: U, O and R.

Unidirectional (U) mode is used either for a simplex channel, or in the initialisation stage for a duplex channel. In this mode, periodic refreshes of the PDU header information are mandatory, to allow the decompressor to reset itself ready to receive further data.

After a period of time the compressor assumes that the decompressor is fully initialised, so transitions from the initialisation state to first-order state.

Bidirectional Optimistic (O) and Bidirectional Reliable (R) modes both make use of a back channel to provide feedback on the quality of the link so the transmitter can make better assumptions about the state that the
receiver requires. Although both O and R modes provide a better compres­
sion rate than U mode, they are inappropriate for broadcast digital radio
systems as there is no native back-channel.

2.4 Summary

In chapter 1 the fields of interest for this research were outlined. In this chapter
have looked at different digital radio standards, some with specific applications
and some designed to be more general purpose. We have seen some error pro­
tection techniques suitable for increasing reliability in broadcast data and some
mechanisms to reduce the overhead of transmission, which may be implemented in
real standards. This chapter has also provided an in depth understanding of data
service protocols over DAB, DMB and DVB-h, and on this knowledge chapter 3
goes on to develop both models and a simulation representing these datacasting
standards. The results of the models and simulation are presented in chapter 4
and the conclusions of this work are drawn together in chapter 5.
Chapter 3

Performance Modelling

Chapter 1 outlined the fields of interests for this research. The digital broadcast mechanisms DAB, DMB and DVB-h were outlined in chapter 2.

In this chapter, the mathematical modelling and the software algorithms used to describe these standards are developed. The results from these models and simulations are presented in chapter 4, and the conclusions drawn in chapter 5.

3.1 Network Quality

In section 2.2.3 various network constructions and the metrics that can be applied to them were discussed.

A common metric used is the ratio of payload to total transmitted data, usually called efficiency. The typical description of efficiency is given in equation 3.1.

$$E = \frac{P_t}{P_t + O}$$ (3.1)

where:

- $E$ is the transmission efficiency
- $P_t$ is the total payload transmitted
- $O$ is the total overhead transmitted

Properly, efficiency is a description of transmission, not of a channel. For the efficiency to be a valid channel description, the channel must be either lossless or very nearly lossless. This is common in wired networks, especially switched wired networks, like 10baseT, 100baseTX and 1000baseT twisted pair Ethernet, or fibre optic Ethernet based computer networks.

Wireless radio networks cannot guarantee the kind of reliability offered by their wired network equivalent technologies, due to the greatly-increased probability of
interference in a received message and/or a lack of coverage in the radio field. The quality of reception is a description of a data channel:

$$Q = \frac{P_r}{P_t k}$$

(3.2)

where:

$Q$ is the reception quality

$P_r$ is the total payload received correctly. This does not include repeated data.

For example: if the same 4 packets ($a$, $b$, $c$ and $d$) were sent twice, and the receiver gathered ‘a c d a’ then $P_r$ would equal the sum of the sizes of $a$, $c$ and $d$, not the sum of the sizes of $2a$, $c$ and $2d$.

$k$ is the number of times the payload is transmitted.

Neither traditional efficiency, nor the quality of reception metric, give the full picture of a channel. How good a channel is should be based on what is placed into the channel, and the reliability of the data which is taken out of the channel. For this work, a combined metric has been developed, and is known throughout this document as performance.

$$C = EQ$$

$$= \frac{P_t}{P_t + O} \times \frac{P_r}{P_t k}$$

$$= \frac{P_t P_r}{(O + P_t) P_t k}$$

$$= \frac{P_r}{k (O + P_t)}$$

(3.3)

where:

$C$ is the channel performance

The channel performance metric given in equation 3.3 makes the assumption that if there is one error, all of the transmission has to be retransmitted to gather the missing data. This is a valid assumption for simplex channels, where the only form of reliability can be gained by repeated retransmissions of the same data [67].

DAB, DMB and DVB-h all provide a mechanism for repeating part of their carousel more often than once a carousel. In duplex channels carrying a connection-oriented protocol like TCP, the receiver is able to inform the transmitter that a specific part of the transmission is missing, greatly reducing the amount of data that is repeated by the transmitter.
Even with a retransmission, there is no guarantee of the correct receipt of the data. In both the carousel and the connection-oriented model, all this mechanism does is improve the probability of receipt. Because the retransmission is of a smaller part of the data, that smaller part of the data has a greater likelihood of success than is modelled in the simplex mechanism.

This does mean that this formula can be generalised, as follows:

\[
C = \frac{\sum_{i=0}^{n-1} P_{ri}}{\sum_{i=0}^{n-1} P_{ti}} \times \frac{\sum_{i=0}^{n-1} P_{ti}}{\sum_{i=0}^{n-1} k_i (O_i + P_{ti})}
\]

\[= \frac{\sum_{i=0}^{n-1} P_{ri}}{\sum_{i=0}^{n-1} k_i (O_i + P_{ti})} \tag{3.4}\]

Where:

- \(n\) is the number of PDUs
- \(P_{ti}\) is the transmitted payload for the PDU \(i\)
- \(P_{ri}\) is the received payload for the PDU \(i\)
- \(O_i\) is the transmitted overhead for the PDU \(i\)
- \(k_i\) is number of retransmissions of the PDU \(i\)

We now compare the channel performance metric with Efficiency, Throughput and Goodput, which were discussed in section 2.2.3. The equations for Throughput and Goodput can be seen in equations 3.5 and 3.6.

\[
t = B \left( \frac{P_t}{P_t + O} \right) \tag{3.5}\]

\[
g = B \left( \frac{P_r}{k(O + P_t)} \right) \tag{3.6}\]

Where, in addition to the parameters we have already seen:

- \(B\) is the theoretical maximum bandwidth of the channel, in bits per second
- \(t\) is the throughput of the channel, in bits per second
- \(g\) is the goodput of the channel, in bits per second
When the equations for throughput and efficiency are compared, it is apparent that they are strongly related: efficiency is measured between 0 and 1, where throughput is measured in bits per second. This difference means that efficiency may be used to compare two channels with different bandwidths, indicating which makes better use of the channel available, where the throughput metric shows the time in an absolute fashion.

A similar comparison may be made with goodput and channel performance: the same difference applies, goodput is measured in bits per second, and channel performance as a value between 0 and 1. Similarly, goodput would be used where an absolute comparison of bandwidth is necessary, where as channel performance is employed when comparing the overhead and loss over channels with differing bandwidths.

When implementing a network, it is desirable to understand the perceived bandwidth for the user, and goodput excels in this application. The strength of channel performance is as a development tool for a new network protocols, or for contrasting different existing protocols for best likelihood of performance. The removal of time from the metric allows unlike network configurations may be directly compared.

3.2 Modelling MSC SPM

Section 2.3.2 reviewed the MSC SPM standard. This mechanism is employed by many parts of the DAB data broadcasting mechanism, and is one possible carrier for a TDC. This section provides:

- a mathematical model of the probability of correct receipt of such a TDC
- a mathematical model of the time that such a TDC would take to be transmitted
- an enhanced mathematical model of the probability of correct receipt of such a TDC
- an enhanced mathematical model of the time that such a TDC would take to be transmitted

3.2.1 Mathematical Model of Probability of Correct Receipt

The following assumptions are made in the construction of this model:
• The channel is affected by errors in a constant Uniform Random Bit Error Rate (BER) pattern

• There is only one size of MSC SPM packet for each transmission, and these are not affected by the size of the logical frame carrying them

• The packets may only be sent once. If there is a caching mechanism provided by the protocol being carried in the TDC, it is beyond the scope of this model

Equation 3.7 models the probability of the correct receipt of all of the data sent for one MSC SPM packet. It is likely that a number of packets may be required to carry the PDU of the next protocol in the stack.

\[ P_{\text{packet}} = (1 - x)^b \]  \hspace{1cm} (3.7)

Where:

- \( x \) is the uniform random Bit Error Rate (BER), where: \( x \in \mathbb{R} : 0 \leq x \leq 1 \)
- \( b \) is the number of bits in a packet, where: \( b \in \{192, 384, 576, 768\} \)

Equation 3.8 presents the model for a PDU which does not fit integrally into one MSC SPM packet.

\[ P_{\text{packets}} = (1 - x)^{bp} \]  \hspace{1cm} (3.8)

\[ p = \left\lfloor \frac{8S}{b - 40} \right\rfloor \]  \hspace{1cm} (3.9)

Where, in addition to the variables already presented:

- \( S \) is the size, in bytes, of the PDU of the next protocol
- \( p \) is the number of MSC SPM packets required to transmit the PDU of the next protocol.

3.2.2 Mathematical Model of Time Taken

Using the assumptions and variables that we have seen already in section 3.2.1, equation 3.10 presents the model for time taken for a given transmission.

\[ T = \frac{bp}{B} \]  \hspace{1cm} (3.10)

Where, in addition to the variables already presented:
$T$ is the time taken for the transmission, in seconds

$B$ is the bandwidth available to the transmission, in bits per second

### 3.2.3 Enhanced Mathematical Model of Probability of Correct Receipt

This section presents an enhanced mathematical model of a TDC over MSC SPM. The following assumptions are made in the construction of this model:

- The channel is affected by a constant Uniform Random Bit Error Rate (BER)
- There may be many different sizes of MSC SPM packet, determined by what the protocols above it in the stack, and the logical frame, require
- The packets may only be sent once. If there is a caching mechanism provided by the protocol being carried in the TDC, it is beyond the scope of this model

Equation 3.11 models the probability of correct receipt of all of the data sent over an MSC SPM TDC where the MSC SPM packets may be of different sizes.

$$p_{\text{packets}} = 1 - \prod_{P=1}^{P} (1 - x)^{b_P}$$

(3.11)

Where, in addition to the parameters already seen:

- $b_P$ is the number of bits in the $P^{th}$ packet, where: $b_P \in \{192, 384, 576, 768\}$
- $P$ is the index used to reference a specific MSC SPM packet

### 3.2.4 Enhanced Mathematical Model of Time Taken

To accompany the enhanced model of probability presented in section 3.2.3, here and enhanced model of time is presented in equation 3.12.

$$T = \frac{\sum_{P=1}^{P} b_P}{B}$$

(3.12)

### 3.3 Modelling MSC Datagroups over MSC SPM

We reviewed the mechanics of sending MSC Datagroups over MSC SPM in section 2.3.4. This section builds on the model presented in section 3.2 to additionally
model MSC Datagroups, which may be used to carry another form of TDC. This section provides:

- a mathematical model of the probability of correct receipt of such a TDC, which builds on the model presented in section 3.2.1
- a mathematical model of the time that such a TDC would take to be transmitted, which builds on the model presented in section 3.2.2
- a mathematical model of the minimum number of repeated transmissions of the set of MSC Datagroups on a carousel to provide a given likelihood of all of the transmitted data arriving unaffected by error
- an enhanced mathematical model of the probability of correct receipt of such a TDC, which builds on the model presented in section 3.2.3
- an enhanced mathematical model of the time that such a TDC would take to be transmitted, which builds on the model presented in section 3.2.4
- There is one object
  - The object is split equally into datagroups
  - A datagroup may be cached for use in a later carousel rotation
  - A FEC frame may only contain whole datagroups
  - One SPM packet may contain all or part of one datagroup
  - All SPM packets are the same size
  - The channel is affected by a uniform random Bit Error Rate (BER)
  - To receive the data, an entire carousel rotation is necessary

### 3.3.1 Mathematical Model of Probability of Correct Receipt

In the construction of this model, the following assumptions are made beyond those already outlined in section 2.3.4:

- in this model, only the MSC SPM mechanism is employed
- all MSC Datagroups are all uniform in size
- all MSC Datagroups will be transmitted the same number of times
The model of the probability of the receipt of all data sent with no error is shown in equation 3.13. The model of this form of TDC includes a data repetition mechanism, as the MSC Datagroup mechanism includes native support for datagroup repetition.

\[
P_{\text{repeatedPackets}} = 1 - \left(1 - (1 - x)^p\right)^k
\]  

(3.13)

Where, in addition to the variables already presented in section 3.2:

- \( p \) is redefined to be the number of MSC SPM packets required to transmit one MSC Datagroup.
- \( D \) is redefined to be the total number of bytes in the MSC Datagroup.
- \( k \) is the number of times the MSC Datagroups are transmitted.

Equation 3.13 models the probability of the correct receipt of all of the data sent for one MSC Datagroup. It is likely that a number of packets may be required to carry the PDU of the next protocol in the stack. Equation 3.14 presents the probability of receipt for a model for a PDU which does not fit integrally into one MSC Datagroup. In this model, we additionally assume that all of the MSC Datagroups are identical in size.

\[
R = \left(1 - \left(1 - (1 - x)^p\right)^k\right)^d
\]  

(3.14)

Where, in addition to the variables already presented:

- \( d \) is the number of MSC Datagroups to be modelled.

3.3.2 Mathematical Model of Time Taken

Using the assumptions and variables that we have seen already in section 3.3.1, equation 3.15 presents the model for time taken for a given transmission.

\[
T = \frac{dkbp}{B}
\]  

(3.15)

3.3.3 Mathematical Model To Find TheMinimum Number Of Carousel Rotations Required

We have already seen the model to determine the likelihood of all of the data transmitted arriving unaffected by error, in equation 3.14. This has been rearranged to provide the mechanism to discover the minimum number of retransmissions.
required to meet a specified likelihood of the correct receipt of all of the data, in equation 3.16.

In addition to the assumptions used in section 3.3.1, this model assumes only complete rotations of the carousel may be used.

\[
k \geq \left[ \frac{\log (1 - R^{\frac{1}{d}})}{\log (1 - (1-x)^{bp})} \right]
\]  
(3.16)

Equation 3.16 has been derived from equation 3.14 as follows:

\[
\left( 1 - (1 - (1-x)^{bp})^k \right)^d = R
\]
\[
1 - (1 - (1-x)^{bp})^k = R^{\frac{1}{d}}
\]
\[
(1 - (1-x)^{bp})^k = 1 - R^{\frac{1}{d}}
\]
\[
k \log \left( 1 - (1-x)^{bp} \right) = \log \left( 1 - R^{\frac{1}{d}} \right)
\]
\[
k = \frac{\log \left( 1 - R^{\frac{1}{d}} \right)}{\log \left( 1 - (1-x)^{bp} \right)}
\]
\[
k \geq \left[ \frac{\log \left( 1 - R^{\frac{1}{d}} \right)}{\log \left( 1 - (1-x)^{bp} \right)} \right]
\]

### 3.3.4 Enhanced Mathematical Model of Probability of Correct Receipt

This section presents an enhanced mathematical model of a TDC over MSC Datagroups over MSC SPM. The following assumptions are made in the construction of this model

- in this model, only the MSC SPM mechanism is employed
- the MSC SPM packets may vary in size, dependant on the MSC datagroup and the logical frame
- all MSC Datagroups are all uniform in size
- there is no requirement for the MSC Datagroups to be sent the same number of times as each other

Equation 3.17 presents the probability of correct receipt of an object transmitted with a range of differently sized MSC Datagroups, each potentially requiring a
different number of retransmissions, and each of the Datagroup retransmissions potentially requiring different sized MSC SPM packets.

\[ R = \prod_{D=1}^{d} \left(1 - \prod_{K=1}^{k} \left(1 - \prod_{P=1}^{p} (1 - z)^{b_{PK}}\right)\right) \]  (3.17)

Where, in addition to the parameters already seen:

- \( b_{PK} \) is the number of bits in the \( P^{th} \) packet of the \( D^{th} \) MSC Datagroup on its \( K^{th} \) transmission, where: \( b_{PK} \in \{192, 384, 576, 768\} \)

- \( K \) is the index used to reference a specific MSC Datagroup iteration

- \( D \) is the index used to reference a specific MSC Datagroup

### 3.3.5 Enhanced Mathematical Model of Time Taken

To accompany the enhanced model of probability presented in equation 3.17, here the model of time taken is presented in equation 3.18.

\[ T = \frac{\sum_{D=1}^{d} \sum_{K=1}^{k} \sum_{P=1}^{p} b_{PKD}}{B} \]  (3.18)

### 3.4 Modelling MSC EPM

Section 2.3.3 reviewed the MSC EPM standard, the mechanism of which is employed by many parts of the DAB data broadcasting mechanism, and is one possible carrier for a TDC. This section provides:

- a mathematical model of the probability of correct receipt of such a TDC
- a mathematical model of the time that such a TDC would take to be transmitted
- an enhanced mathematical model of the probability of correct receipt of such a TDC

### 3.4.1 Mathematical Model of Probability of Correct Receipt

As we saw when we reviewed the MSC EPM mechanism in section 2.3.3, the MSC EPM mechanism works by surrounding the MSC SPM mechanism with
the virtual FEC frame, which appends nine RS packets to the selection of MSC SPM packets which are included in the frame. This mechanism makes it difficult to mathematically model an individual MSC SPM packet, which is where the payload is.

This section models the probability of the entire FEC frame arriving either entirely unaffected by error, or with some bytes in error, but within the tolerances that the RS (204,188) mechanism can correct, and can be seen in equation 3.19.

$$F = \left( \sum_{i=0}^{8} \binom{204}{i} s^i (1-s)^{204-i} \right)^{12}$$

(3.19)

Where:

$F$ is the MSC EPM FEC frame error rate

$s$ is the symbol error rate, where: $s = 1 - (1 - x)^8$ because the symbol is one byte

$x$ is the uniform random Bit Error Rate (BER), where: $x \in \{R : 0 \leq x \leq 1\}$

204 is the total number of bytes sent in one RS (204, 188) FEC packet

8 is the maximum number of bytes that may arrive in error in one RS (204, 188) FEC packet

12 is the number of rows in the MSC EPM FEC frame

### 3.4.2 Mathematical Model of Time

Although the MSC EPM FEC frame is fixed in length, and thus the time taken to transmit an MSC EPM FEC frame must also be fixed, the time taken to receive the payload contained within one MSC EPM FEC frame varies depending on error. Equation 3.20 models the time taken to transmit a FEC frame, and the worst case reception time. Equation 3.21 models the best case time taken to receive a FEC frame, where the RS packets may be ignored as the MSC SPM packets all arrived unaffected by error.

$$T_{\text{max}} = \frac{188 \times 12 + 9 \times 24}{B}$$

$$= \frac{2472}{B}$$

(3.20)
\[ T_{\text{min}} = \frac{188 \times 12}{B} = \frac{2256}{B} \] (3.21)

As we have seen, there is no trivial way to create the probability for one MSC SPM packet carried in a MSC EPM channel: the smallest unit whose probability can be calculated uniquely being the FEC frame, so modelling repeating PDUs that do not integrally fit FEC frames is a bad representation. The FEC frame is capable of carrying a number of MSC SPM packets, which we saw in section 2.3.3. Equations 3.22 and 3.23 show the smallest and largest possible payloads of an MSC EPM FEC frame.

\[ S_{\text{min}} = 188 \times 12 - 5 \frac{188 \times 12}{24} = 1786 \] (3.22)
\[ S_{\text{max}} = 188 \times 12 - 5 \left\lceil \frac{188 \times 12}{96} \right\rceil = 2136 \] (3.23)

3.4.3 Enhanced Mathematical Model of Probability of Correct Receipt

Within the MSC SPM packet footer there is a 32 bit CRC field, which is tested to see if the MSC SPM packet is valid. The receiver is at liberty to check these fields on reception, even if the MSC EPM mechanism is employed. It is possible that the receiver will receive an entire application data table free from error, and the corresponding RS data table corrupted by error. In this case, the receiver will have already received the MSC SPM packets correctly and used them. This is modelled in equation 3.24.

\[ F = \left( \sum_{i=0}^{8} \binom{204}{i} s^{i} (1 - s)^{204-i} \right)^{12} + \left( \sum_{j=9}^{16} \binom{16}{j} s^{j} (1 - s)^{204-j} \right)^{12} \] (3.24)

3.5 Modelling MSC Datagroups over MSC EPM

As we have seen in section 3.4, the assumptions for modelling MSC EPM are far more stringent than were listed for the equivalent MSC SPM model, which we saw in section 3.3.

This section provides:

* a mathematical model of the probability of correct receipt of MSC Datagroups over an MSC EPM channel
• a mathematical model of the time that MSC Datagroups take over such a channel

• an enhanced mathematical model of the probability of correct receipt of MSC Datagroups

3.5.1 Mathematical Model of Probability of Correct Receipt

As we saw in section 2.3.3, the MSC EPM standard implements the FEC frame. Each FEC frame may contain more than one datagroup, which we saw in section 2.3.4. As MSC Datagroups may be cached, it becomes hard to mathematically model MSC Datagroups spanning more than one FEC frame, especially if a frame contains parts of more than one MSC datagroup. Therefore, in this model we assume that one FEC frame may contain one MSC Datagroup.

The assumptions made for the model of DAB EPM are as follows:

• There is one object

• The object is split equally into datagroups

• A datagroup may be cached for use in a later carousel rotation

• A FEC frame may only contain whole datagroups

• One SPM packet may contain all or part of one datagroup

• All SPM packets are the same size

• The channel is affected by a uniform random Bit Error Rate (BER)

• To receive the data, an entire carousel rotation is necessary

We have already seen the model of MSC EPM FEC frames in section 3.4.1. Here we build on this model to provide the mechanism for MSC Datagroups over MSC EPM. Equation 3.25 models the number of FEC frames required to transport one MSC Datagroup.

\[
f = \left\lfloor \frac{bp}{188 \times 12 \times 8} \right\rfloor = \left\lfloor \frac{bp}{18048} \right\rfloor
\]  

(3.25)

Using the number of frames variable, seen in equation 3.25, and the model for obtaining the probability of correct receipt of a FEC frame, which we saw in
section 3.4, we can now build the model for MSC Datagroups over MSC EPM. The probability of receipt for a cached object in this model can be seen in equation 3.26.

\[ R = \left( 1 - (1 - F^f)^k \right)^d \]  \hspace{1cm} (3.26)

### 3.5.2 Mathematical Model of Time Taken

Building on the model for the time it takes an MSC EPM FEC frame to be transmitted, which we saw in section 3.4.2, the model for the time it takes to transmit the MSC Datagroups can be seen in equation 3.27.

\[ t = fT \]  \hspace{1cm} (3.27)

Where:

- \textit{f} is from equation 3.25
- \textit{T} is from either equation 3.20 or 3.21, for the maximum and minimum time taken respectively

### 3.5.3 Enhanced Mathematical Model of Probability of Correct Receipt

It is possible to improve the datagroup size model, obtaining the number of frames in the manner modelled in equation 3.28.

\[ f = \frac{\sum_{P=1}^{PD} b_{PD}}{188 \times 12 \times 8} = \frac{\sum_{P=1}^{PD} b_{PD}}{18048} \]  \hspace{1cm} (3.28)

This model takes into account the effect of varying MSC SPM packet sizes due to restrictions provided by the logical frame and the MSC Datagroup which it is carrying.

### 3.6 Modelling MPEG2-TS over MSC Stream Mode

We reviewed the mechanism for transmitting MPEG2-TS over MSC Stream Mode in section 2.3.6. In this review, we saw that the MSC Stream Mode mechanism relies on both bit-wise adjustment, to dissipate the energy required for transmission and reception; and also a Forney approach convolutional encoding mechanism. Both of these aspects of the standard will not be implemented in this model as
neither will have an effect on the probability in the context of a channel affected by a consistent Uniform Random Bit Error Rate (BER).

This section now provides:

- a mathematical model of the probability of correct receipt of an MPEG2-TS channel over MSC Stream Mode
- a mathematical model of the time that an MPEG2-TS channel take over such a channel

3.6.1 Mathematical Model of Probability of Correct Receipt

The model of the probability of the receipt of one MPEG2-TS packet arriving over an MSC Stream Mode channel unaffected by error is is shown in equation 3.29.

\[ M = \sum_{i=0}^{8} \binom{204}{i} s^i (1 - s)^{204-i} \]  
(3.29)

Where:

- \( M \) is the MPEG2-TS packet error rate
- \( s \) is the symbol error rate, where: \( s = 1 - (1 - x)^8 \)
- \( x \) is the uniform random BER, where: \( x \in \{ \mathbb{R} : 0 \leq x \leq 1 \} \)

It is reasonable to expect a number of MPEG2-TS packets to be used to convey data. Building on the model given in equation 3.29, equation 3.30 models the probability of a number of MPEG2-TS packets arriving correctly.

\[ P_{stream} = M^p \]  
(3.30)

\[ p = \frac{S}{188} \]  
(3.31)

Where, in addition to the variables already defined in this section:

- \( p \) is the number of MPEG2-TS packets required for the transmission
- \( S \) is the size of the data to be sent over the MPEG2-TS channel
3.6.2 Mathematical Model of Time Taken

The queueing delay caused by the Forney approach interleaver was addressed in section 2.3.6, where it was defined to be 204 bytes. Although this is a delay, it does not affect the bandwidth, as the delay is constant for all bytes, so the effect of the Forney Interleaver is a constant in this model. The model for time taken is shown in equation 3.32.

\[ T = \frac{8(204 + 204p)}{B} \]  

Where, in addition to the variables already defined in section 3.6.1:

- \( B \) is the bandwidth in bits per second
- \( T \) is the time taken in seconds

3.7 Modelling IP over MPE-FEC over DVB-h

We reviewed the mechanism for transmitting data using the MPE-FEC over DVB-h channel in section 2.3.7. The following are presented in this section:

- a mathematical model of the probability of correct receipt of data over an MPE-FEC over DVB-h channel
- a mathematical model of the time that such a channel would take to be transmitted
- an enhanced mathematical model of the probability of correct receipt of such a TDC

3.7.1 Mathematical Model of Probability of Correct Receipt

A similar problem exists when addressing modelling MPE-FEC over DVB-h to the problem of modelling MSC Datagroups over MSC EPM, which we saw in section 3.5: it is non-trivial to model an individual PDU within the frame. The assumptions made in the construction of the models in this section are:

- one MPEG2-TS frame may contain part or all of a single IP packet
- all IP packets are uniform in size
- the channel is affected by a Uniform Random BER
• there is only one object on the carousel
• there is never a need to transport an MPEG2-TS null packet
• the RS data table is not punctured, which was reviewed in section 2.3.7. Therefore any 32 bytes in error for any MPE-FEC frame row may be corrected
• the application data table does not employ the padding columns outlined in section 2.3.7
• the impact of the MPE packet header is ignored, as it has no impact on the default RS mechanism used across the MPE-FEC frame

The model of the probability of the correct receipt of for an individual MPE-FEC frame can be seen in equation 3.33.

\[ F = \left( \sum_{j=0}^{32} \binom{255}{j} s^i (1 - s)^{255-i} \right)^r \]  

(3.33)

Where:

- \( F \) is the probability of an MPE-FEC frame arriving correctly
- \( r \) is the number of rows in the MPE-FEC frame
- \( s \) is the symbol error rate, where: \( s = 1 - (1 - x)^8 \)
- \( k \) is the number of times the MSC Datagroups are transmitted
- \( x \) is the uniform random BER, where: \( x \in \{ R : 0 \leq x \leq 1 \} \)

Building on the model of the probability of the correct receipt of a single MPE-FEC frame, as seen in equation 3.33, we can see the model for a number of IP packets transmitted using MPE-FEC over DVB-h in equation 3.34.

\[ R = \left( 1 - (1 - F^k) \right)^i \]  

(3.34)

\[ i = \left\lfloor \frac{D}{fp} \right\rfloor \]  

(3.35)

\[ f = \left\lfloor \frac{191r}{p} \right\rfloor \]  

(3.36)

Where, in addition to the parameters we have already seen in this section:
\[ p \] is the size of the IP packets

\[ f \] is the number of IP packets carried in one application data table

\[ i \] is the number of MPE-FEC frames required for the transmission

\[ D \] is the total size of the data to be sent

### 3.7.2 Mathematical Model of Time Taken

The model of time taken for an IP over MPE-FEC over DVB-h is shown in equation 3.37. This model is based on the MPE-FEC standard mechanisms described in section 2.3.7.

\[
T = \frac{188 \times \left[ \frac{(13+p) \times \left[ \frac{64}{5} \right]}{184} \right] + 188 \times \left[ \frac{f(16+p)}{184} \right]}{B}
\]

(3.37)

### 3.8 Software Simulation of the Standards

There are complications in the broadcast standard that require a less generalised model than can reasonably be mathematically constructed. PDU size may be influenced by constraints imposed by protocols both above and below the layer in the network stack, for example the size of MSC SPM packets in an MSC EPM transmission are influenced by the logical frame, which is influenced by the MSC SPM and MSC EPM packets; and the MSC SPM packets in an MSC EPM channel are affected by the size of MSC Datagroup size. This can be modelled, but when constructed the model becomes very specific to that configuration. A simulator is able to be more flexible in configuration, allowing for alternative experiment configurations to be trivially implemented. The model was designed to use a Uniform Random BER as this will normally give a worst-case performance of the error protection mechanisms, and may be used to compare the simulator to the model trivially, to verify that the simulator is performing correctly.

This section will present:

- The design for the software simulator
- The system structure to make best use of resources

### 3.8.1 Design of the Software Simulator

Every protocol has a unique set of rules for conversing between network end-nodes. Many of these rules may be generalised:
• Keepability

Non-keepable protocols do not have enough addressing information for the protocol to cache each PDU for the case of retransmission. The pseudo code can be found in algorithm 3 for the mathematical model, and algorithm 1 for the simulation algorithm.

Keepable protocols have enough addressing information for each PDU to cache the PDU for the case of retransmission. The pseudo code can be found in algorithm 4 for the mathematical model, and algorithm 2 for the simulation algorithm.

• Tested for error

Non-checksummed protocols do not check every PDU for the impact of error. The pseudo code can be found in algorithm 5 for the simulation algorithm.

Checksummed protocols check every PDU for the impact of error, and should a PDU be found to have arrived in error, the PDU is dropped. The pseudo code can be found in algorithm 6 for the simulation algorithm.

• Corrected from error

Uncorrected protocols transmit the data normally. The pseudo code can be found in algorithm 9 for the mathematical model, and algorithm 7 for the simulation algorithm.

Error corrected protocols transmit additional information alongside the data, which the receiver can then apply to the received data. The pseudo code can be found in algorithm 8 for the simulation algorithm. There are two examples of correction functions, which can be found in algorithms 10 and 11.

• The number of unique PDUs carried from the higher layer

Single carrying protocols have PDUs that may contain part or all of no more than one PDU of the protocol above this one in the network stack. The pseudo code can be found in algorithm 14 for the mathematical model, and algorithm 12 for the simulation algorithm.

Multiple carrying protocols have PDUs that may contain part or all of any number of PDUs of the protocol above this one in the network stack. The pseudo code can be found in algorithm 15 for the mathematical model, and algorithm 13 for the simulation algorithm.
Table 3.1 presents the protocols to be simulated in relation to the attributes as listed above. It is clear that there are many possible combinations of these attributes, which are necessary for different protocols. An ideal design would implement these attributes from many places, so that the specific code for each protocol has to do the minimum, allowing for easier testing procedures. For this, the object-oriented technique of multiple-inheritance for functions was employed.

Each protocol will inherit the functions for the different functionalities from abstract classes.

As these rules are in all protocols in some combinations, definition fits the object-oriented concept of multiple-inheritance for functions well.

<table>
<thead>
<tr>
<th>Keepable</th>
<th>Checksummed</th>
<th>Error Corrected</th>
<th>Single Carrying</th>
<th>Multiple Carrying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carousel data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DAB EPM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>DAB SPM</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datagroup</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical Frames</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTP</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Streamed Data</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Protocols And Their Attributes

**Algorithm 1:** unKeepable:haveSeenPDUBefore(PDU.label, data)

Input: PDU.label, the identifier for a specific PDU
Input: data, the data symbols received
Output: data, the data symbols received as it is irrelevant whether the PDU was found before

begin
| return data;
end
Algorithm 2: haveSeenPDUBefore(PDU.label, data)

Input: PDU.label, the identifier for a specific PDU
Input: data, the data symbols received
Output: data, the corrected data symbols if this PDU was received correctly before, or the data symbols as received

begin
    if lookup(PDU.label) returns found then
        return makeAllCorrect(data);
    if anyErrors(data) then
        return data;
    lookupUpdate(PDU.label);
    return data;
end

Algorithm 3: mathematicaIRepetion(probability, transmissions)

Input: probability, probability of reception of this PDU
Output: Probability after transmissions

begin
    return probability;
end

Algorithm 4: Keepable:mathematicaIRepetion(probability, transmissions)

Input: probability, probability of reception of this PDU
Input: transmissions, the number of transmissions this PDU will experience
Output: Probability of correct receipt after transmissions

begin
    return 1-(1-probability)^transmissions;
end

Algorithm 5: nonChecksummed:performChecksum(data)

Input: data, the data symbols received
Output: data, the data symbols received as no checksum is performed

begin
    return data;
end
Algorithm 6: Checksummed: performChecksum(data)
Input: data, the data symbols received
Output: data, the data symbols all marked as errant, if any errors were present
begin
  if anyErrors(data) then
    return setAllToError(data);
  return data;
end

Algorithm 7: unCorrected: correct(data)
Input: data, the data symbols received
Output: data, the data symbols received as no correction is performed
begin
  return data;
end

Algorithm 8: Corrected: correct(data)
Input: data, the data symbols received
Output: data, the data symbols having had the correction mechanism applied
begin
  return particularErrorCorrectionAlgorithm(data);
end

Algorithm 9: unCorrected: correctMathematically(probability)
Input: probability, probability of reception of this PDU
Output: probability, as no correction mechanism is applied
begin
  return probability;
end

Algorithm 10: EPM: correctMathematically(probability)
Input: probability, probability of reception of this PDU
Output: Resultant probability of correct receipt after correction mechanism is applied
begin
  s = probability / (188*12+24*9);
  result = 0;
  for i=0; i<8; i=i+1 do
    result = result + choose(i) * (1-s)^204-i * s^i;
  result = result^12;
  return result;
end
Algorithm 11: MPEG2TS:correctMathematically(probability)

Input: probability, probability of reception of this PDU
Output: Resultant probability of correct receipt after correction mechanism is applied

begin
  \[ s = \text{probability} / (188 \cdot 12 + 24 \cdot 9); \]
  for \( i = 0; i \leq 8; i = i + 1 \) do
    \[ \text{result} = \text{result} + \text{choose}(i) \times (1 - s)^{204 - i} \times s^i; \]
return result;
end

Algorithm 12: SingleCarrier:transmit(PDU_label, dataSize)

Input: PDU_label, identifier of the PDU from the layer above in the stack
Input: dataSize, size in bytes of the data to be transmitted
Output: data, the stream of data received by this layer

begin
  data = \emptyset;
  while \( \text{dataSize} > 0 \) do
    newdata = sublayer \rightarrow \text{transmit}(\text{maxPossiblePDUSize}(\text{dataSize}));
    newdata = \text{correct}(\text{newdata});
    newdata = \text{haveSeenPDBefore}(\text{concat}(\text{PDU_label}, \text{iteration}), \text{newdata});
    newdata = \text{performChecksum}(\text{newdata});
    concat(data, newdata);
    \[ \text{dataSize} = \text{dataSize} - \text{sizeof}(\text{newdata}); \]
return(data);
end
Algorithm 13: MultipleCarrier:transmit(PDU_label, dataSize)

Input: PDU_label, identifier of the PDU from the layer above in the stack
Input: dataSize, size in bytes of the data to be transmitted
Output: dataToReturn, the stream of data received by this layer

begin
    dataToReturn = ∅;
    while dataSize > 0 do
        if dataSize < sizeLeftInThisPDU then
            sizeLeftInThisPDU = sizeLeftInThisPDU - dataSize;
            (tmp, data) = split(data, dataSize);
            concat(dataToReturn, tmp);
            dataSize = 0;
        else
            concat(dataToReturn, data);
            dataSize = dataSize - sizeLeftInThisPDU;
            sizeLeftInThisPDU = 0;
            data = sublayer → transmit(maxPossiblePDUSize);
            data = correct(data);
            data = haveSeenPDUBefore(concat(label, iteration), data);
            data = performChecksum(data);
            sizeLeftInThisPDU = sizeof(data);
        end
    end
    return(dataToReturn);
end

Algorithm 14: SingleCarrier:findProbability(dataSize, rotations)

Input: dataSize, size in bytes of the data to be transmitted
Input: rotations, number of rotations of the carousel
Output: probability, the probability of correct receipt of the PDU

begin
    probability = 1;
    while dataSize > 0 do
        dataToSendThisTime = maxPossiblePDUSize(dataSize);
        dataSize = dataSize - dataToSendThisTime;
        thisResult = sublayer → findProbability(dataToSendThisTime);
        thisResult = correctMathematically(thisResult);
        thisResult = mathematicalRepetition(thisResult, rotations);
        probability = probability * thisResult;
    end
    return probability;
end
Algorithm 15: MultipleCarrier:findProbability(dataSize, rotations)

Input: dataSize, size in bytes of the data to be transmitted
Input: rotations, number of rotations of the carousel
Output: probability, the probability of correct receipt of the PDU

begin
    probability = 1;
    while dataSize > 0 do
        if dataSize < sizeLeftInThisPDU then
            dataSize = 0;
        else
            dataToSendThisTime = maxPossiblePDUSize();
            dataSize = dataSize – dataToSendThisTime;
            thisResult =
            sublayer →findProbability(dataToSendThisTime);
            thisResult = correctMathematically(thisResult);
            thisResult = mathematicalRepetition(thisResult, rotations);
            probability = probability * thisResult;
        end
    end
return probability;
end
3.8.2 Design of the System

Within the system there are a number of distinct jobs that can be segregated:

**Insert new requests** into the database. This process first checks to see if the job has been performed previously. If there are experiments that have already been run, it uses the results of those experiments. If the experiment is a new configuration, then the experiment is inserted and marked ready to be run.

**Run Pending Experiments** as and when there are experiments waiting. It is likely that a user may wish to insert many experiments at one time. This stage may use a network of distributed computers to perform many experiments concurrently.

**User Interface** needs to be accessed by users on different operating systems. A simple solution is to implement the user interface as a web application, so that the only requirements that the user must have is a web browser and a network connection capable of reaching the web server.

The user may want to do a combination of three tasks:

- **Create a new plot** for a comparison of a network, or some networks, that has not been performed so far
- **Retrieve a plot** for a comparison that was previously created
- **Check the status of the system** to ensure that each of parts of the system are performing correctly

The system components and their interaction with each other is shown in figure 3.1.

**DRBE Database** is implemented using Mysql on a Debian GNU/Linux platform. Mysql was chosen as it is lightweight, free and easy to configure. The entity relationship diagram can be seen in figure B.1.

All of the other components need access to the database, so the DBI communication mechanism was as it is also lightweight, and easy to implement across a network.

**Inserter Daemon** performs two roles: managing new experiment requests; and organising which experiments need to be run.

When a user wishes to add a new experiment, the inserter daemon takes this experiment, checks the database to find the points that have already been inserted, performs the mathematical modelling of any points that haven't yet been inserted, and stores the experiment construction.
Periodically, the inserter daemon checks the database to discover what experiments are pending, and if it finds any it queues them ready for a process daemon. When a process daemon requests a job, assuming that there are experiments pending, the inserter daemon provides the process daemon with the job identification, and marks that experiment as being performed by that process daemon. On job completion, the process daemon informs the inserter daemon that the job has been completed, and the inserter daemon forgets the relationship between the experiment and process daemon.

The Inserter Daemon was written in perl as this is a fast language to develop as there are many libraries freely available. It can also be fast to run, as many of the libraries come as binary executables, so the overhead of using an interpreted language is lessened.

Process Daemon performs the simulation for each iteration of the experiment. It polls the Inserter Daemon periodically for an experiment, and when it is allocated a job, it collects the required settings from the DRBE database, and runs the experiment. When the experiment has been completed, the results are placed into the DRBE database, and the Process Daemon requests another job from the Inserter Daemon.

There are typically many experiments required to be analysed, so there can be many Process Daemons concurrently running. The farm of machines running Process Daemons was constructed to increase the speed of the experiment completion. Typically six machines were used, although as other machines became available the farm grew to twelve machines.

The Process Daemon was written in perl, as this is a fast language to develop as there are many libraries freely available, and for ease of interoperability with the Inserter Daemon. It can also be fast to run, as many of the libraries come as binary executables, so the overhead of using an interpreted language is lessened.

Web User Interface was developed so that any user on any system could access it. It is build using CGI/perl within Apache on a Debian GNU/Linux system.

3.9 Object size Analysis

There are a number of different sized objects which can be expected to be sent over a broadcast digital radio channel. To produce a useful working model for the
Once the farm machine has been allocated a job, it discovers the settings from the database.

Once the farm machine has completed its job, it places this experiment iteration's answers into the database.

The mathematical modeling is performed and the experiment is placed into the database.

The inserter daemon periodically checks the database for pending jobs.

The process daemon(s) poll the inserter daemon for pending experiments.

Valid experiment is sent to the inserter daemon.

The status of the inserter daemon is checked.

The status of all the farm machines is checked.

Web user sets up an experiment.

Web user retrieves answers to a completed experiment.

Web user checks system status.

Figure 3.1: The Simulation System Structure
channels, it is necessary to have an understanding of the object sizes which are likely to be transmitted.

3.9.1 MPEG Streams

An MPEG video stream is likely to be used across all of the standards, and will therefore be an easy comparison.

MPEG video streams consist of three frame types: I, P and B. These frames can be expected to be transmitted in a repeating pattern of: I B B P B P B P B B P. Example sizes for these frames can be seen in table 3.2.

<table>
<thead>
<tr>
<th>Frame</th>
<th>Size (kilobytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>5015</td>
</tr>
<tr>
<td>P</td>
<td>1446</td>
</tr>
<tr>
<td>B</td>
<td>1256</td>
</tr>
</tbody>
</table>

Table 3.2: Sizes of frames in an MPEG video stream [39]

3.10 Error Profiling

The mathematical models presented in this chapter are capable of modelling Uniform Random BERs. It is important for the simulator to have a Uniform Random BER to compare the results of the model with the simulation results. Such a profile is obtained using a pseudo random number generator.

The Uniform Random BER profile will model a worst-case error performance of a channel. To gain a clearer understanding of how the channels will perform in practice it is necessary to capture some real data. Real capture DAB data was not available, so an experiment was developed in conjunction with Arqiva to provide this data. Arqiva provided four sub-channels, known as the all-zero channels, on the Cambridge Experimental Ensemble, with the configuration shown in table 3.3. The four all-zero channels were configured to transmit bits at state 0 only. This transmission option was chosen for three reasons: the transmitter could perform this transmission natively; errors would be easy to find as all the bits should be in state 0; and if a more complicated bit sequence were used, synchronising the captured data with the expected data would be hard. Cambridge was chosen as the location as Arqiva have an experiment and development ensemble at Cambridge, which they had space on for this experiment.

Arqiva lent a specialist DAB receiver system, consisting of: receiver hardware capable of providing statistics on the reception; a laptop for logging; and a Global
Positioning System (GPS) receiver. This system captured three main pieces of information: GPS location; the raw data received, including any errors; and a calculated Uniform Random BER value for the reception. It is the raw data that is of particular interest to this research, as the simulator can take this and indicate how different protocols would have performed on the same channel at the same time.

The specialist DAB receiver can only capture one sub-channel at a time, so to capture the four all-zero channels, the experiment needed to be performed four times: once for each all-zero channel. The specialist DAB receiver system was mounted in a car and driven on a route around Cambridge on the route as shown in figure 3.2 [27]. This route consisted of all fast roads, and was chosen as the likelihood was that the atmospheric and traffic conditions would be fairly consistent over the length of time to perform the four trips required to capture the four all-zero channels. Arqiva also knew that the reception quality on the road near the transmitter was very good, whilst the hilly terrain at the east-most extent of the route would mean a very low likelihood of any reception, so the route would display the characteristics of both good and bad reception.

<table>
<thead>
<tr>
<th>Channel Name</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Zeros 1</td>
<td>8 kbps</td>
</tr>
<tr>
<td>All Zeros 2</td>
<td>128 kbps</td>
</tr>
<tr>
<td>All Zeros 3</td>
<td>96 kbps</td>
</tr>
<tr>
<td>All Zeros 4</td>
<td>384 kbps</td>
</tr>
</tbody>
</table>

Table 3.3: Cambridge All Zero Channel Configuration
Figure 3.2: The Route of capture of the all-zeros data sets [27]
Chapter 4

Evaluation and Results

In chapter 2 DAB, DMB, and DVB-h, were described. Chapter 3 described both a mathematical model and a software simulation, of these standards. This chapter now looks at the results of the mathematical model and the software simulation, to assess the performance of real channels.

This chapter also sets out to prove best practice, choice and configuration for as much of the soft part of the protocol as possible, drawing on the understanding of Quality of Service (QoS) techniques provided in section 2.2.3; this is to provide knowledge to help select the most appropriate network stack, and then tune the chosen network stack to provide the most suitable channel. This chapter provides an understanding of:

- the performance metric
- the mathematical models presented in chapter 3
- the enhanced models that provide a mechanism to check the simulator against, also presented in chapter 3
- the results of the simulation

In doing so we see the best configuration for each standard, and a direct comparison of the standards against each other.

Later, in chapter 5, the implications of this new knowledge is assessed.

4.1 Performance Metric

The performance metric was introduced in section 3.1, along with the values $E$, $Q$ and $C$. Here the three metrics are contrasted to provide an understanding of when each of the metrics are useful. This understanding can then be applied to
the rest of this chapter to assess the performance of the standards, and to be able to compare them directly.

The parameters involved in this section are:

\( C \) is the channel performance

\( E \) is the transmission efficiency

\( k \) is the number of times the payload is transmitted.

\( O \) is the total overhead transmitted

\( P_r \) is the total payload received correctly. This does not include repeated data.

\( P_t \) is the total payload transmitted

\( Q \) is the reception quality

There are a number of assumptions that are made to produce this section.

- This set up represents no existing protocol. It is to demonstrate the formulae only.

- There is only one object carried in one generic PDU sent in the transmission for this experiment.

To understand the metric, a set of hypothetical channels will be depicted. These do not relate to any existing channel, they have been designed to display the effect of classes of channel on the metric.

4.1.1 Performance of a reliable channel, including transmission of overhead

In this section the effect of overhead is presented. This experiment represents a reliable channel carrying a protocol. It can be seen in figure 4.1 that \( E \) and \( C \) have the same values. and that \( Q \) is unaffected by overhead. This is because overhead only effects throughput of a channel, and this is only a concern at the time of transmission.

The trend implies that large PDUs are best as the ratio of overhead to payload is small, increasing the efficiency of the transmission: loss has no affect on the channel as this is a channel unaffected by error, so the efficiency is the only factor to consider.
4.1.2 Performance of a lossy channel with reliable reception after two transmissions, including transmission of overhead

In this section the channel is affected by error, which has the effect of requiring one additional carousel transmission. After two transmissions, all of the transmitted data is received. It can be seen in figure 4.2 that \( E \) and \( C \) have the same shape, with \( C \) being half of \( E \). This is due to \( Q \) being 50%.

The trend implies that big PDUs are best, which is the same trend as was previously seen in section 4.1.1. This indicates that, although loss now affects this channel, the efficiency of the channel is still the predominant factor on the channel.

4.1.3 Performance of a lossy channel after two transmissions, including transmission of overhead

In this section the effect of losing about half the data transmitted, in addition to the effects displayed in section 4.1.2.

- Reliable reception
Figure 4.2: Reliable Performance after two transmissions, including transmission of overhead

- 10 bytes of overhead
- 2 transmissions

A number of observations can be made:

- $E$ continues to increase in the same manner as previously seen
- $Q$ at best achieves 25%, which is a reflection that two cycles are needed (50% quality) and at most, half the data arrives correctly
- $C$ increase as the PDU payload size increases, but with some backsliding where less than half of the data arrives correctly.

The trend seen in figure 4.3 implies that big PDUs are generally best. Although there may be some cases where the PDU payload size performs less well than a smaller one, this seems to be a negligible effect, and may be ignored.

### 4.1.4 Performance of a lossy channel needing increasing transmissions for reliable reception, including transmission of overhead

- Reliable reception
Figure 4.3: Lossy Performance after two transmissions, including transmission of overhead

- 10 bytes of overhead
- 2+payload transmitted transmissions

The trend implies that there is a point where performance peaks. This is due to the increasing requirement for more carousel rotations as the PDU payload size increases. This result is in contrast to the results previously found in this section, where the PDU payload size was always improved as it increased.

It seems that tuning the protocol may be necessary in cases where the carousel repetition count may increase.

4.2 Employing the Mathematical Model

To understand the performance of any network it is useful to understand the performance of each PDU in the network stack, therefore this section of the research begins with analysing the performance of datagroups and SPM packets. Later in this section, the implication of this new knowledge will be applied to channels carrying data.

In chapter 3 some models were developed to represent the data broadcasting standards commonly found in DAB, DMB, DAB+, and DVB-h. Here we build on
the understanding provided in section 4.1 to analyse the formulae to discover the best set up for the standards. As with all models, some assumptions are made:

- The channel is affected by uniform random bit errors
- All of the PDUs for a protocol are the same size
- A PDU is never restricted to a size by a lower protocol in the stack
- There is only ever one object to be carried
- The bandwidth is constant

4.2.1 MSC SPM

In this section we see the probability and time taken for stream broadcast data over an MSC SPM TDC as modelled in sections 3.2.1 and 3.2.2. We will go on to provide tuning configuration information, as well as a comparison of MSC SPM with MSC EPM in section 4.2.5.

This section presents both the performance of an individual PDU for a range of PDU sizes, over an MSC SPM TDC, and a quantity of data segregated into a
range of PDU sizes sent over an MSC SPM TDC. Table 4.1 shows the settings that will be used in this section.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAB SPM Packet Payloads</td>
<td>∈ {19, 43, 67, 91} Bytes</td>
</tr>
<tr>
<td>Generic PDU sizes</td>
<td>∈ {N : 1 \leq n \leq 10000} Bytes</td>
</tr>
</tbody>
</table>

Table 4.1: Possible variables for a datagroups over MSC SPM network stack

**TDC PDUs over MSC SPM**

As was discussed in section 3.2, the assumed use for a TDC over MSC SPM is for streamed data. Although protocols may make use of this type of TDC and implement a cache mechanism of its own, we assume in this section that this channel is for streamed data only. Later we will go on to analyse the performance of cached data in section 4.2.3. Figure 4.5 shows the probability of a single MSC SPM packet arriving when it is carried in an MSC SPM channel affected by a range of Uniform Random BERs. Figure 4.5 shows that the best likelihood that all of the packet arrives is achieved by using smaller packets. We now shall look
at other metrics to analyse the channel to better understand the implications of the BER on the channel.

Figure 4.6 shows the probability of the correct receipt of all of the data transmitted against the time it takes to transmit. There is a distinct step pattern displayed on all the lines in figure 4.6. This corresponds to the size of the MSC SPM packet size used in that channel. The step shows where an additional MSC SPM packet is required to transmit the data being sent. Some discrete values have been taken from this model, and are displayed in table 4.2. It can be seen from the results in table 4.2 that the trend is towards small quantities of data being send over this channel for the greatest reliability.

The object sizes 19, 43, 67 and 91 were selected to fit in integrally one MSC SPM packet for each of the valid MSC SPM packet sizes, and it can be seen that there the highest probability of arrival for a 24 byte MSC SPM packet, with a decreasing likelihood of all of the MSC SPM packet arriving as the packets sizes increase. The value 188 was chosen as it is the size of an MPEG2-TS packet, and the value 1,000,000 was chosen as it represents a larger stream of data.

In the case of the 188 byte object, representing the MPEG2-TS packet, the best likelihood of correct arrival is obtained by choosing the 72 byte MSC SPM packet option. This is because the simple model only provides utilisation of uniformly
<table>
<thead>
<tr>
<th>Size of PDU (bytes)</th>
<th>Packet Size (bytes)</th>
<th>Probability (%)</th>
<th>Time Taken (seconds)</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>24</td>
<td>98.10</td>
<td>0.0005</td>
<td>79.17</td>
</tr>
<tr>
<td>43</td>
<td>48</td>
<td>96.23</td>
<td>0.001</td>
<td>89.58</td>
</tr>
<tr>
<td>67</td>
<td>72</td>
<td>94.40</td>
<td>0.0015</td>
<td>0</td>
</tr>
<tr>
<td>91</td>
<td>96</td>
<td>92.61</td>
<td>0.002</td>
<td>0</td>
</tr>
<tr>
<td>188</td>
<td>24</td>
<td>82.53</td>
<td>0.005</td>
<td>0</td>
</tr>
<tr>
<td>188</td>
<td>48</td>
<td>82.53</td>
<td>0.005</td>
<td>0</td>
</tr>
<tr>
<td>188</td>
<td>72</td>
<td>84.13</td>
<td>0.0045</td>
<td>0</td>
</tr>
<tr>
<td>188</td>
<td>96</td>
<td>79.42</td>
<td>0.006</td>
<td>0</td>
</tr>
<tr>
<td>1,000,000</td>
<td>24</td>
<td>0.00</td>
<td>26.316</td>
<td>0</td>
</tr>
<tr>
<td>1,000,000</td>
<td>48</td>
<td>0.00</td>
<td>23.256</td>
<td>0</td>
</tr>
<tr>
<td>1,000,000</td>
<td>72</td>
<td>0.00</td>
<td>22.389</td>
<td>0</td>
</tr>
<tr>
<td>1,000,000</td>
<td>96</td>
<td>0.00</td>
<td>21.98</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.2: Probability of the correct arrival, the Time Taken and the Channel Performance for a range of PDU sizes sent over a 384 kb/s MSC SPM TDC, affected by a Uniform Random BER of 0.0001

sized MSC SPM packets. Both the 72 and the 96 byte cases require 3 packets to be transmitted, as can be seen in equation 4.1, and that means more bits are required to be sent using the 96 byte option.

\[
\begin{bmatrix}
188 \\
96 - 5
\end{bmatrix} = \begin{bmatrix}
188 \\
72 - 5
\end{bmatrix}
\]

(4.1)

In all of the other cases the larger the MSC SPM packet, the more likely the data will be received correctly. This is attributed to the decrease in overhead at the MSC SPM layer.

This model provides the probability of all of the data arriving, which is different from the quantity of data that was correctly received, so when we model performance we must take a minimum accepted likelihood for all of the data being received correctly. We shall assume that 95% likelihood is the minimum acceptable likelihood of arrival. Figure 4.11 shows the performance of this channel. Although the lines for the 72 and 96 byte MSC SPM packet models are present, the likelihood of the data arriving over this channel is always less than the desired minimum probability of 95%, so the performance has been set to 0. The 24 and 48 byte MSC SPM packet models are present. These both have two distinct steps: the little steps show the performance of at each transmitted byte increment, indicating a better transmission efficiency as the fixed size MSC SPM packet become closer to being fully utilised; the large steps indicate where a new MSC SPM packet was required; and the drop to 0 shows where the reception quality has fallen to below
Figure 4.7: The Channel Performance of a range of PDU sizes sent over a 384 kb/s MSC SPM TDC

the desired 95% likelihood,

TDC Data over MSC SPM

Earlier in this section we came to an understanding of the performance of a single PDU sent over an MSC SPM TDC. This section now addresses the effect of the size of PDU when transmitting larger objects. For this exercise, a 1,000,000 byte object was chosen. This was chosen as it represents a stream with enough length to represent a human noticeable time period of streamed data.

Figure 4.8 shows the probability of correctly receiving all of a 1,000,000 byte object over a channel affected by a Uniform Random BER of $10^{-4}$. It can be seen in figure 4.8 that there is no configuration combination which can provide a greater than negligible probability of correct reception. This means that the performance of the channel, using this model, is also negligible at all points.
Figure 4.8: The Probability of Correct Receipt and the Time Taken for a 1,000,000 byte object transmitted over a range of PDU sizes over a 384 kb/s MSC SPM TDC
4.2.2 MSC EPM

In this section we see the probability and time taken for stream broadcast data over an MSC EPM TDC as modelled in sections 3.4.1 and 3.4.2. Later, in section 4.11, we will provide a tuned configuration and a comparison with MSC SPM.

This section now presents both the performance of an individual PDU for a range of PDU sizes, over an MSC EPM TDC, and a quantity of data segregated into a range of PDU sizes sent over an MSC EPM TDC. Table 4.3 shows the settings that will be used in this section.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAB SPM Packet Payloads</td>
<td>( \in {19, 43, 67, 91} ) Bytes</td>
</tr>
<tr>
<td>Generic PDU sizes</td>
<td>( \in {N : 1 \leq n \leq 10000} ) Bytes</td>
</tr>
</tbody>
</table>

Table 4.3: Possible variables for a datagroups over MSC EPM network stack

TDC PDUs over MSC EPM

As discussed in section 3.4, this model does not find the probability of an individual MSC SPM packet sent over an MSC EPM channel, but it does find the probability of an individual MSC EPM FEC frame. Because a FEC frame is of uniform size, regardless of the use of the MSC SPM packets used to send the data over the channel, the FEC frame error rate is the same for all channels, as can be seen in figure 4.9. Across all lines the lowest probability of correct reception of a single MSC EPM FEC frame is 0.999999999998031, which has been rounded to 1 on the scale. Although the performance is degrading, it is not degrading so much that it is likely to cause data loss at the receiver.

Now other channel performance analysis is performed on MSC EPM channels. Figure 4.10 shows the probability of the correct receipt of all of the data transmitted against the time it takes to transmit. The probability of all of the sizes of MSC SPM packets in the MSC EPM channel have a probability of arriving correctly so close to 100% that the rounding error in the code used to implement the mode has rounded to 100%, as can be seen in figure 4.10. The time taken goes up in large steps, which correspond to the requirement of an additional MSC EPM FEC frame to contain all of the MSC SPM packets. Some discrete values have been taken from the model, and may be seen in table 4.4. It is clear from table 4.4 that the RS FEC mechanism, as modelled, is strong enough to correct errors which impacted the similar TDC, as we saw in section 4.2. Figure 4.11 shows the performance of data over an MSC EPM TDC. As we saw in section 4.2.1, we assume that all of the data on the channel arrives if the likelihood of arrival is equal to
Figure 4.9: The Effect of Uniform Random BERs on the probability of MSC SPM packets carried in an MSC EPM channel

Figure 4.10: The Probability of Correct Receipt and the Time Taken for a range of PDU sizes over a 384 kb/s MSC EPM TDC
Table 4.4: Probability of the correct arrival, the Time Taken and the Channel Performance for a range of PDU sizes sent over a 384 kb/s MSC EPM TDC, affected by a Uniform Random BER of 0.0001

<table>
<thead>
<tr>
<th>Size of PDU (bytes)</th>
<th>Packet Size (bytes)</th>
<th>Probability (%)</th>
<th>Time Taken (seconds)</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>24</td>
<td>100.00</td>
<td>0.05</td>
<td>0.77</td>
</tr>
<tr>
<td>43</td>
<td>48</td>
<td>100.00</td>
<td>0.05</td>
<td>1.74</td>
</tr>
<tr>
<td>67</td>
<td>72</td>
<td>100.00</td>
<td>0.05</td>
<td>2.71</td>
</tr>
<tr>
<td>91</td>
<td>96</td>
<td>100.00</td>
<td>0.05</td>
<td>3.68</td>
</tr>
<tr>
<td>188</td>
<td>24</td>
<td>100.00</td>
<td>0.05</td>
<td>7.61</td>
</tr>
<tr>
<td>188</td>
<td>48</td>
<td>100.00</td>
<td>0.05</td>
<td>7.61</td>
</tr>
<tr>
<td>188</td>
<td>72</td>
<td>100.00</td>
<td>0.05</td>
<td>7.61</td>
</tr>
<tr>
<td>188</td>
<td>96</td>
<td>100.00</td>
<td>0.05</td>
<td>7.61</td>
</tr>
<tr>
<td>1,000,000</td>
<td>24</td>
<td>100.00</td>
<td>28.84</td>
<td>72.24</td>
</tr>
<tr>
<td>1,000,000</td>
<td>48</td>
<td>100.00</td>
<td>25.49</td>
<td>81.72</td>
</tr>
<tr>
<td>1,000,000</td>
<td>72</td>
<td>100.00</td>
<td>24.57</td>
<td>84.81</td>
</tr>
<tr>
<td>1,000,000</td>
<td>96</td>
<td>100.00</td>
<td>24.10</td>
<td>86.44</td>
</tr>
</tbody>
</table>

Figure 4.11: The Channel Performance of a range of PDU sizes sent over a 384 kb/s MSC EPM TDC
or greater than the desired 95%. In the case of the MSC EPM channel, however, this may be ignored as the likelihood is close to 100% at all points.

All four lines follow the same trend, which is the utilisation of the MPE EPM FEC frame. The four lines, do diverge at the points where the overhead of the MSC SPM packets requires an additional FEC frame, and it is the MSC SPM packet overhead which also prematurely caps the upper bound of the saw tooth in each of the four cases.

**TDC Data over MSC EPM**

Earlier in this section we came to an understanding of the performance of a single PDU sent over an MSC EPM TDC. This section now addresses the effect of the size of PDU when transmitting larger objects. For this exercise, a 1,000,000 byte object was chosen. This is the same length as was chosen in section 4.2.1, and was chosen again to provide a direct comparison of the results.

Figure 4.12 shows the probability of correctly receiving all of a 1,000,000 byte object over a channel affected by a Uniform Random BER of $10^{-4}$, and the time that it took to transmit the object. It can be seen in figure 4.12 that there is no configuration combination that reduces the probability of correct receipt from
near-certain. The time that it takes to transmit all of the 1,000,000 byte object reduces in time as the PDU size increases. The performance of this channel can be seen in figure 4.13. It is clear from the channel performance metric for this

![Channel Performance Chart](image)

Figure 4.13: The Channel Performance of a 1,000,000 byte object transmitted over a range of PDU sizes sent over a 384 kb/s MSC EPM TDC channel that the larger the PDU size performs best.
4.2.3 Datagroups over MSC SPM

In this section we build on the understanding derived from the results presented in section 4.2.1 to understand the performance of a TDC over MSC Datagroups over MSC SPM is presented, using the models presented in sections 3.3.1 and 3.3.2.

Performance of the Datagroup and SPM PDUs

The possible values of datagroups and SPM packets are presented in table 4.5.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datagroup Payload</td>
<td>1 - 8,189 bytes</td>
</tr>
<tr>
<td>DAB SPM Payload</td>
<td>∈ {19, 43, 67, 91} Bytes</td>
</tr>
<tr>
<td>Channel BER</td>
<td>1 - 0</td>
</tr>
</tbody>
</table>

Table 4.5: Possible variables for a datagroup over MSC SPM network stack

Figure 4.14 shows the entire curve for a transmission of exactly one datagroup and all four SPM packet sizes. It assumes a reliable channel.

A sawtooth formation can be seen in figure 4.14, and the four lines tend to a stable state by datagroup payload size of 8,189 bytes.
The lines are separated by the natural amount of overhead, the amount of overhead that exists when the datagroup size is utilising the SPM packets perfectly. This is listed in tables A.5, A.6, A.7 and A.8.

The sawtooth formation, which can be seen more clearly in figure 4.15, is due to the ratio of datagroup size to SPM packet payload. The SPM packet must carry padding if it is not full utilised. As the datagroup size increases it fits the SPM packet more efficiently, the amount of padding needed to make the packet size is reduced. This is the upward slope of the sawtooth. The fall in the sawtooth formation is the moment when the datagroup size has increased such that the SPM packets no longer fit the datagroup pattern well, so the amount of needed padding is dramatically increased.

The little steps that construct the saw teeth formations are due to using discrete points on the chart.

![Graph of Reliable Channel Zoomed In (1 Object per Datagroup)](image)

**Figure 4.15: A closer view of the saw teeth**

Analysing the results in figure 4.14, the trend appears to be that large SPM packets and large datagroups are best, with the optimum settings are:

- Datagroup payload size should be 8,088 bytes
- DAB SPM packet size should be 96 bytes

Figure 4.16 shows the results of the same configuration as displayed in figure 4.14,
but this time the channel is affected by a uniform random BER of $10^{-4}$, defined as the worst BER that audio services may continue to function.

It is understood that the transmission should provide at least a 95% likelihood of 100% of the data being received correctly.

It is assumed that if the content did not have at least a 95% likelihood of complete arrival after 20 cycles of the carousel, that it would never be received.

![Figure 4.16: Performance of the entire datagroup range, given a channel affected by a uniform random BER of $10^{-4}$](image)

The effect of error on the channel is obvious when comparing the results in figures 4.16 and 4.14. The effect on performance attributed to a need to repeat the transmission. It must be assumed that if there was an error in reception that the entire carousel needs to be received. The dramatic drop relates to one additional carousel rotation.

The graph shows a drop to 0%, where the datagroup sizes are between 2,000 and 3,000 bytes, which indicates when the transmission needed more than 20 carousel cycles to gain at least 95% likelihood of complete arrival.

Analysing the results in figure 4.16, the trend appears to be toward large SPM packets, but small datagroups.

To better understand the performance, the results presented in figure 4.16 are presented again in figure 4.17 with the scales altered.

Analysing the results in figure 4.16, the trend appears to be toward large SPM...
Figure 4.17: Performance of datagroup payload sizes of 10 – 500 bytes, given a $10^{-4}$ Uniform Random BER packets, but small datagroups.

**Performance of Objects over Datagroups over MSC SPM**

At this point in the research, the effect of datagroups and SPM packets are understood. It is now important to understand the effect that this has on realistic object sizes.

Figure 4.18 shows the entire curve for a fixed object size of 1,000,000 bytes, with all datagroup payload sizes and all four SPM packet sizes. It assumes a reliable channel.

Comparing figure 4.18 with figure 4.14, it can be noted that the sawtooth pattern in figure 4.18 alters direction as the datagroup payload size increases. This is due to a change of the importance in the ratio of SPM packet, datagroup payload and object size. When the datagroup payload size is small, the overhead that the datagroup introduces is most significant. When the datagroup payload size is large, the effect miss matching the object with the datagroup payload becomes most significant. This displays a limitation with this simulation, as the datagroup standard permits transmitting the correct size required. This model assumes that the last datagroup contains padding.

Figure 4.19 portrays the effect of a uniform random BER of $10^{-6}$. 

99
Figure 4.18: Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a reliable channel.

The graph in figure 4.19, there are artefacts where the performance alters by a carousel cycle for a short duration. It is assumed that a whole carousel transmission is needed, and that equates to a number of bytes that must be used. As the datagroup payload size increases, and the relationship between the datagroup and SPM packet alters, the amount of padding changes, sometimes making the transmission more efficient. The artefact happens when the transmission is more efficient, so the transmission needs less bytes, which leads to a whole carousel less being needed.

As with figure 4.19, the results shown in figure 4.20 clearly show the artefacts. It is assumed that a whole carousel transmission is needed, and that equates to a number of bytes that must be used. As the datagroup payload size increases, and the relationship between the datagroup and SPM packet alters, the amount of padding changes, sometimes making the transmission more efficient. The artefact happens when the transmission is more efficient, so the transmission needs less bytes, which leads to a whole carousel less being needed.

Figure 4.21 shows the impact of a Uniform Random BER of $10^{-5}$. The effect of the increased BER is a requirement of an increased number of carousel cycles to maintain a minimum of 95% likelihood of complete arrival of
Figure 4.19: Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-6}$.

Figure 4.22 shows the impact of a Uniform Random BER of $10^{-4}$, which is the expected worst case BER that should effect the channel, according to the design mandate of DAB. It can be seen from figure 4.22 that at the expected worst case BER for the channel that datagroup payload sizes of less than 1,000 bytes are required. Figure 4.23 shows this more clearly. From figure 4.23 it can be concluded that big SPM packets sizes with one datagroup PDU designed to fit exactly in one SPM packet performs best.

Table 4.6 shows the optimum settings for objects over datagroups over MSC SPM, given the constraints of this mathematical model.

<table>
<thead>
<tr>
<th>SPM Packet Size (bytes)</th>
<th>Datagroup payload size (bytes)</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>65</td>
<td>13.54</td>
</tr>
<tr>
<td>48</td>
<td>75</td>
<td>15.62</td>
</tr>
<tr>
<td>72</td>
<td>56</td>
<td>15.55</td>
</tr>
<tr>
<td>96</td>
<td>80</td>
<td>16.67</td>
</tr>
</tbody>
</table>

Table 4.6: Optimum Settings for datagroups for a given SPM packet size, derived from the simple mathematical model of a datagroups over MSC SPM channel.
Figure 4.20: A close up of an carousel rotation artefact, caused by the rounding in this model.

Figure 4.21: Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-5}$.
Figure 4.22: Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-4}$. 
Figure 4.23: Improved view of the effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over a datagroups over MSC SPM over a channel affected by a uniform random BER of $10^{-4}$
4.2.4 Datagroups over MSC EPM

MSC EPM was designed to be a backward compatible enhancement to MSC SPM, which would provide an EPM enabled receiver additional error correction information. In this section, equivalent experiments are presented to those found in section 4.2.3.

Performance of the PDUs

The datagroups over MSC EPM network stack is similar to the datagroups over MSC SPM network stack. The difference is the addition of the EPM FEC frames which contain RS error correction. The EPM FEC frame structure are inserted into the packet mode transmission after the SPM packets, which was described in section 2.3.3.

Figure 4.24 depicts the effect of differing datagroup payloads over an EPM channel.

![Figure 4.24: Performance of the entire datagroup range](image)

It can be seen that there is a very clear sawtooth formation. The upward trend depicts the performance as the datagroup fills the application table. The sharp fall is due to another EPM FEC frame being required to transmit the datagroup.

The trend derived from figure 4.24 is that big SPM packets and big datagroups are best, at sizes that exactly fit the EPM FEC frame(s) carrying them,
The theoretical worst-case BER for a DAB channel is $10^{-4}$. The performance of EPM over a channel with this loss rate is depicted in figure 4.25.

![Figure 4.25: Performance of the entire datagroup range, given a channel affected by a uniform random BER of $10^{-4}$](image)

It should be noted that when comparing the results presented in figures 4.24 and 4.25, the results are identical. This is because the level of correction provided by the RS encoding fixes all errors in this channel.

There are 204 bytes in one row of a FEC frame, therefore there are 1,632 bits. The RS encoding used is RS (204:188:8), which means that any 8 byte errors can be corrected. The theoretical worst case BER on a DAB channel is $10^{-4}$ or 1 in 10,000. In a FEC frame there are 25,792 bits, which means that it is reasonable to expect 5 bit errors spread across 2 FEC frames. This is well within the correction capabilities of the FEC frame.

The trend derived from figure 4.25 is that big SPM packets and big datagroups are best, at sizes that exactly fit the EPM FEC frame(s) carrying them.

**Performance of Objects over Datagroups over MSC EPM**

There is merit in understanding the effect that an object will experience when being carried over datagroups over MSC EPM.

Figure 4.26 depicts the performance of a 1,000,000 byte object carried over datagroups over MSC EPM. The large sawtooth pattern witnessed in figures 4.24
Figure 4.26: Effect of varying datagroup payload size on the performance of a 1,000,000 byte object transmitted over datagroups over MSC EPM over a reliable channel

and 4.25, which was caused by the waste of an EPM FEC frame, no longer has a significant impact on the performance.

The trend presented in figure 4.26 is for big SPM packets and big datagroups.

The effect of carrying data over the worst-case BER channel is presented in figure 4.27, from which it can be seen that performance is the same as those previously in figure 4.26.

Table 4.7 shows the optimum settings for objects over datagroups over MSC EPM, given the constraints of this mathematical model. The trend found is for big SPM packets and big datagroups.

<table>
<thead>
<tr>
<th>SPM Packet Size (bytes)</th>
<th>Datagroup payload size (bytes)</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>6,411</td>
<td>72.11</td>
</tr>
<tr>
<td>48</td>
<td>4,762</td>
<td>81.56</td>
</tr>
<tr>
<td>72</td>
<td>3,473</td>
<td>84.63</td>
</tr>
<tr>
<td>96</td>
<td>4,630</td>
<td>86.25</td>
</tr>
</tbody>
</table>

Table 4.7: Optimum Settings for DAB EPM, derived from the simple mathematical model
4.2.5 The Comparison of the TDCs

This section contrasts the four TDCs presented thus far, with particular interest on the affect of FEC mechanisms and repetition on the channel performance.

The Comparison of MSC SPM with MSC EPM

Figure 4.28 compares the packet probability of correct reception of the four MSC SPM packet sizes over an MSC SPM channel with the frame error rate of the four MSC SPM packet sizes sent over an MSC EPM channel. While this is not the ideal comparison, as the probability of correct reception for the MSC EPM frame has been penalised for containing more MSC SPM packet error rate, it can be seen in figure 4.28 MSC EPM still compares very favourably against an MSC SPM channel, as the model of the probability of correct MSC EPM finds that the probability is nearly one for one FEC frame to arrive correctly.

When comparing the channel performance results presented in sections 4.2.1 and 4.2.2, it is apparent that the additional overhead of the RS packets in the MSC EPM mechanism have an effect on the length of time a transmission takes. The worst case comparison is between the single 24 byte MSC SPM packet where
Figure 4.28: The Comparison of the Channel Performance of an MSC SPM TDC with an MSC EPM TDC, for a Range of PDU Sizes
the MSC EPM mechanism takes 100 times longer. This assumes that an entire MSC EPM FEC frame must be used to carry one MSC SPM packet. A better comparison to concentrate on is the case of the 1,000,000 byte object, where the difference is still large at 2.5 seconds, but the time taken is still in the same order of magnitude.

The comparison of the channel performance can be seen in figure 4.29. The comparison of the performance shows a stark contrast between the two MSC packet mode mechanisms: in the case of the models of 24 and 48 byte MSC SPM packets carried in an MSC SPM channel, which can be seen in figures 4.29a and 4.29b, indicates a narrow setting where there is a relatively good performance, below 50 byte payloads. The same data sent over an MSC EPM channel indicates a higher performance at high level of payloads.

The case of the models of 72 and 96 byte MSC SPM packets carried in an MSC EPM channel, which can be seen in figures 4.29c and 4.29d, indicate that the likelihood that no data will be received over an MSC SPM mechanism, again in stark contrast to the same data sent over an MSC EPM channel, where again the better performances tend to the larger units of data sent over the channel.

At this point we may conclude that the tuned configuration for a TDC is either:
• to use MSC SPM with 24 byte MSC SPM packets, and a PDU size of either 19 or 38, to fit in exactly one or two 24 byte MSC SPM packets

• to use MSC SPM with 48 byte MSC SPM packets, and a PDU size of 43 bytes, to fit in exactly one 48 byte MSC SPM packet

• to use 96 byte MSC SPM packets over MSC EPM with a large PDU, greater than 2,000 bytes, to gain the best performance

When comparing the results from the 1,000,000 byte object, there is no case where the MSC SPM performs better than 0, so for the purposes of the comparison, we shall only use the case of the PDU. It is worth remembering that this model can only predict the probability of all of the data arriving. The probability of receiving two PDUs of the same size, on a channel affected by a Uniform Random BER, is the same. This means that it is worth tuning the MSC SPM mechanism for the best PDU performance.
The Comparison of MSC Datagroups over MSC SPM with MSC Datagroups over MSC EPM

Figure 4.30 shows the channel performance of a single datagroup sent using both a TDC consisting of MSC Datagroups/MSC SPM and a TDC consisting of MSC Datagroups/MSC EPM. Comparing the results that we have seen in figures 4.29 and 4.30, it can be seen that there is a distinct benefit in repeated transmissions of the same data on a channel affected by a high BER, where there is no FEC mechanism. It can also be seen that there is no obvious benefit of data repetition within an MSC EPM channel, when using this model. Comparing the MSC SPM and MSC EPM channels in figure 4.30 shows that the best way to protect a channel is to use a FEC mechanism, as this has a smaller affect on the channel performance than repeating the data. These results are confirmed in figure 4.31 where the same channels are compared, this time sending a 1,000,000 byte object. There is no configuration where the channel performance is good concurrently for the MSC SPM based TDCs and the MSC EPM based TDCs.

At this point in the research, it is clear that using a FEC mechanism like that included in MSC EPM provides a clear benefit to a channels performance, where
Figure 4.31: The Comparison of the Channel Performance of a 1,000,000 byte object sent over an MSC Datagram/MSC SPM TDC with a 1,000,000 byte object sent over an MSC Datagram/MSC EPM TDC, for a Range of PDU Sizes.
the underlying channel is affected with a high BER.

4.2.6 IP/MSC Datagroups/MSC EPM for Streamed Data

Most computer networks use IP as the network layer protocol. This has lead to much development of hardware to natively support IP. The broadcast digital radio community has developed a transport standard which incorporates IP, gaining the benefits of cheap hardware.

In this section we see the probability and time taken for stream broadcast data over a streamed data channel utilising IP over MSC Datagroups/MSC EPM; this is the transmission mechanism used in the BT Movio trails. We will go on to provide tuning configuration information, and compare it to MPEG2-TS over Stream Mode, the DMB streamed video mechanism, in section 4.2.8.

This section presents both the performance of an individual PDU, for a range of PDU sizes, and a quantity of data segregated into a range of PDU sizes. Table 4.1 shows the settings that will be used in this section. The IP size is higher than

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP packet sizes</td>
<td>{N : 50 \leq n \leq 8158} Bytes</td>
</tr>
<tr>
<td>MSC Datagroup sizes</td>
<td>{N : 61 \leq n \leq 8178} Bytes</td>
</tr>
<tr>
<td>DAB SPM Packet Payloads</td>
<td>{19, 43, 67, 91} Bytes</td>
</tr>
</tbody>
</table>

Table 4.8: Possible variables for IP over MSC Datagroups over MSC SPM network stack

the standard MTU for IP of 1,500 bytes, as we assume that the broadcaster may choose to use the non-standard jumbo-frames mechanism to convey content from source to transmitter, and therefore it may be desirable to use larger than 1,500 IP packet size. The model used in this section does not cater for the case where one IP packet is sent across two MSC Datagroups. It is unlikely that this will be a significant factor, however, as the IP packet will be fragmented using the standard mechanism [15, 52] and this only results in additional overhead of one IP packet header per MSC Datagroup.

Performance of the IP and SPM PDUs

It can be seen that there is a very clear sawtooth formation. The upward trend depicts the performance as the datagroup fills the application table. The sharp fall is due to another EPM FEC frame being required to transmit the datagroup.

The trend derived from figure 4.32 is that big SPM packets and big datagroups are best, at sizes that exactly fit the EPM FEC frame(s) carrying them,
The theoretical worst-case BER for a DAB channel is $10^{-4}$. The performance of EPM over a channel with this loss rate is depicted in figure 4.33. It should be noted that when comparing the results presented in figures 4.32 and 4.33, the results are identical. This is due to the level of correction provided by the RS encoding, at this level of error the model assumes that all errors in this channel are correctable.

There are 204 bytes in one row of a FEC frame, therefore there are 1,632 bits. The RS encoding used is RS (204:188:8), which means that any 8 byte errors can be corrected. The theoretical worst case BER on a DAB channel is $10^{-4}$ or 1 in 10,000. In a FEC frame there are 25,792 bits, which means that it is reasonable to expect up to 5 bit errors spread across 2 FEC frames. This is well within the correction capabilities of the FEC frame.

The trend derived from figure 4.33 is that big SPM packets and big datagroups are best, at sizes that exactly fit the EPM FEC frame(s) carrying them.

**The Effect of Data on IP/MSC Datagroups/MSC EPM**

Figure 4.34 depicts the performance of a 1,000,000 byte object carried over an IP over datagroups over MSC EPM channel.

The large sawtooth pattern witnessed in figures 4.32 and 4.33, which was caused by the use of an extra EPM FEC frame, no longer has a significant impact on the performance.
Figure 4.33: Performance of the entire datagroup range, given a $10^{-4}$ Uniform Random BER

The trend presented in figure 4.34 is for big MSC SPM packets and big data-groups, and it can be noted that results are identical for a channel affected by a Uniform Random BER of $10^{-4}$.

Table 4.9 shows the smallest found optimum settings for Internet Protocol over Digital Audio Broadcast (DAB-IP) EPM, given the constraints of this mathematical model. The smallest optimum settings are used as it is reasonable to assume that, although this model uses the Uniform Random BER case and may assume that the errors are equally dispersed across the channel, in a real channel the errors will come in bursts. The smaller the packet, the better it’s chances of missing an error burst.

<table>
<thead>
<tr>
<th>SPM Packet Size (bytes)</th>
<th>Datagroup payload size (bytes)</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>5,953</td>
<td>71.85</td>
</tr>
<tr>
<td>48</td>
<td>7,408</td>
<td>81.39</td>
</tr>
<tr>
<td>72</td>
<td>6,667</td>
<td>84.45</td>
</tr>
<tr>
<td>96</td>
<td>7,247</td>
<td>86.07</td>
</tr>
</tbody>
</table>

Table 4.9: Smallest Found Optimum Settings for IP over datagroups over MSC EPM, derived from the simple mathematical model
Figure 4.34: Performance of DAB EPM of a 1,000,000 byte object, given a Reliable Channel

Performance of Objects over RTP over UDP over IP over MSC Datagroups over MSC EPM

Figure 4.35 depicts the performance of a 1,000,000 byte object carried over an RTP over UDP over IP over datagroups over MSC EPM channel.

Comparing the results presented in figure 4.35 with those in figure 4.34, it can be seen that the effect on performance is only slight. This is accounted for by the additional overhead of the RTP and UDP headers.

The results are identical for a channel affected by a uniform random BER of $10^{-4}$.

Table 4.10 provides the best settings for each of the four MSC SPM packet sizes.
Figure 4.35: Performance of one 1,000,000 byte object over RTP over UDP over IP over MSC EPM, given a Reliable Channel

<table>
<thead>
<tr>
<th>SPM Packet Size (bytes)</th>
<th>Datagroup payload size (bytes)</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>7,093</td>
<td>71.73</td>
</tr>
<tr>
<td>48</td>
<td>6,098</td>
<td>81.07</td>
</tr>
<tr>
<td>72</td>
<td>6,850</td>
<td>84.28</td>
</tr>
<tr>
<td>96</td>
<td>5,682</td>
<td>85.71</td>
</tr>
</tbody>
</table>

Table 4.10: Optimum Settings for RTP over UDP over IP over datagroups over MSC EPM, derived from the simple mathematical model
4.2.7 IP/Datagroups/MSC SPM for Streamed Data

As previously seen in section 2.3.3, MSC EPM is technically backward-compatible with MSC SPM. It is conceivable that the BT Movio transmission mechanism, mentioned in section 4.2.6, may be received by an MSC SPM only receiver. This section provides an understanding of how well IP over MSC SPM will perform.

As we saw in section 4.2.1, the probability of streamed data sent over an MSC SPM based channel affected by a BER of $10^{-4}$ is not large enough to be considered viable. The only difference between that TDC/MSC SPM model, and this IP/MSC Datagroups/MSC SPM model is an increase in the overhead required to send the data. We will not analyse this model, as figure 4.8 has already shown that the likelihood of receipt for all possible combinations is 0.

Performance of RTP over UDP over IP over MSC Datagroups over MSC SPM

It is expected that a common protocol to be carried over IP over datagroups over MSC SPM is RTP over UDP, so the performance of the transport layer for real-time streaming applications is analysed here. When the performance of RTP over

![Figure 4.36: Performance of the entire datagroup range](image-url)
the RTP channel has a reduced performance. This is due to the additional 20 bytes of overhead, attributed to the UDP and the RTP packet headers.

A defining difference between the two standards is that RTP is only ever used in a streamed-data context, where IP may be employed to transport carousel data. Figure 4.37 displays the performance of RTP carried over an IP over datagroups over MSC SPM channel affected by a uniform random BER of $10^{-4}$. As can be seen in figure 4.37, the performance for all datagroup sizes is 0. This is because the probability of receiving all of the data sent is statistically less than 95%. If a stream broadcast is made using MSC EPM, it is safe to conclude that MSC SPM only receivers will not receive the data correctly.

### 4.2.8 MPEG2-TS/MSC Stream Mode

The recognised standard mechanism for carrying streamed data over DAB is MPEG2-TS streaming over MSC stream mode [19]. DMB especially utilises this mechanism to carry video data [20].

The transmission mechanism is modelled in section 3.6. We are modelling a channel affected Uniform Random BER, and we assume that the channel is not affected by differences in power consumption at either the transmitter or the
receiver. We also assume that the interleaver has no impact on the Uniform Random profile of the bit errors.

The Effect of PDUs on MPEG2-TS/MSC Stream Mode

Figure 4.38 shows the probability of a single MPEG2-TS packet arriving when it is carried in an MSC Stream Mode channel affected by a range of Uniform Random BERs. It can be seen in figure 4.38 that the likelihood for correct arrival of the

![Graph showing probability of reception for an individual MPEG2-TS packet sent over MSC Stream Mode for a range of Uniform Random BERs.]

Figure 4.38: The Effect of Uniform Random BERs on the probability of MPEG2-TS packets carried in an MSC Stream Mode channel affected by varying

MPEG2-TS packet is very high at all Uniform Random BERs. The theoretical worst-case BER for a DMB channel is $10^{-4}$. The performance of DMB over a channel with this loss rate is depicted in figure 4.40.

It should be noted that when comparing the results presented in figures 4.39 and 4.40, the results are identical. This is because the level of correction provided by the RS encoding fixes all errors in this channel.

The trend derived from figure 4.40 is that big SPM packets and big datagroups are best, at sizes that exactly fit the MPEG2-TS packets carrying them,
Figure 4.39: The performance of a range of PDU sizes sent over an MPEG2-TS/MSC Stream Mode channel

The Effect of Data on MPEG2-TS/MSC Stream Mode

Figure 4.41 depicts the performance of a 1,000,000 byte object carried over a DAB EPM channel. The trend presented in figure 4.41 is for big DAB SPM packets and big datagroups.

The effect of carrying data over the worst-case BER channel is presented in figure 4.42. The results presented in figure 4.42 are the same as those previously seen in figure 4.41.

The trend found is for big PDUs to be carried over MPEG2-TS/MSC Stream Mode.
Figure 4.40: The performance of a range of PDU sizes sent over an MPEGtwoTS/MSC Stream Mode channel affected by a Uniform Random BER of $10^{-4}$.

Figure 4.41: The performance of a 1,000,000 byte object sent over a range of PDU sizes over an MPEGtwoTS/MSC Stream Mode channel.
Figure 4.42: The performance of a 1,000,000 byte object sent over a range of PDU sizes over an MPEG2-TS/MSC Stream Mode channel affected by a Uniform Random BER of $10^{-4}$
4.2.9 Streamed Data over IP/MPE-FEC

MPE-FEC, which we reviewed in section 2.3.7, is a mechanism for the transport of broadcast data that was introduced into the DVB standards with the advent of DVB-h.

As we have already seen in this chapter, the power of using FEC mechanisms has a strong benefit to the channel performance. Although the MPE mechanism exists without the FEC addition, we will not analyse it: because the bandwidth of DVB-h is considerably higher than that of a DAB ensemble [16, 15], and because the standard Uniform Random BER is similar to DAB, we can safely assume that the performance of MPE against MPE-FEC will be equivalent to the comparative performance of MSC SPM against MSC EPM.

The Effect of PDUs on IP/MPE-FEC

So far in this section the size of the PDUs have defined the resolution of the error rate. In the case of IP/MPE-FEC, the resolution is defined by the MPE-FEC frame. The precision on the machine that produced this data for this model was such that the probability was always rounded to 1. This can be ascribed to the powerful RS (255,191) FEC mechanism. The theoretical worst-case BER for a DAB channel is $10^{-4}$. The performance of DVB-h over a channel with this loss
rate is depicted in figure 4.44.

Figure 4.44: Performance of the entire datagroup range, given a $10^{-4}$ Uniform Random BER

It should be noted that when comparing the results presented in figures 4.43 and 4.44, the results are identical. This is because the level of correction provided by the RS encoding fixes all errors in this channel.

The trend derived from figure 4.44 is that big IP packets and few MPE rows are best.

The Effect of Data on IP/MPE-FEC

The effect of carrying a 1,000,000 byte object over the worst-case BER channel is presented in figure 4.45. A distinct step pattern can be seen in figure 4.45. The steps represent where an additional MPE-FEC frame is required to convey the data. The IP packet size being used to carry a section of the 1,000,000 byte object is fixed size, so there is a penalty in performance when the MPE-FEC frame is less well filled by the size of the IP packet.

The trend here is to use an IP packet with between 1,000 bytes and 2,750 bytes of payload.
Performance of a 1M Object over MSC EPM TDC Effected by a $10^{-4}$ Uniform Random Bit Error Rate

Figure 4.45: The performance of a 1,000,000 byte object streamed over a range of PDU sizes over an IP/MPE-FEC/DVB-h affected by a Uniform Random BER of $10^{-4}$

Performance of RTP over UDP over IP over MPE-FEC over DVB-h

Figure 4.46 depicts the performance of a 1,000,000 byte object carried over RTP over UDP over IP over MPE-FEC DVB-h channel.

The trend presented in figure 4.46 is that big PDUs are good, over few MPE-FEC rows.

When the results presented in figure 4.46 are compared to the ones in figure 4.59, the only difference is that the small additional overhead of RTP and UDP have slightly lessened the performance.

The performance results of a channel affected by a uniform random BER of $10^{-4}$ are the same as the results presented of the reliable channel.
Figure 4.46: Performance of one 1,000,000 byte object over RTP over UDP over IP over MPE-FEC DVB-h, given a reliable channel
4.2.10 The Comparison of the Stream Data Standards

We have now used the mathematical model to provide an understanding of the performance of various broadcast streamed data. We now compare the following standards with each other:

- IP/MSC Datagroups/MSC EPM
- MPEG2-TS over MSC Stream Mode
- IP over MPE-FEC over DVB-h

Comparing IP over MSC Datagroups over MSC EPM with MPEG2-TS over MSC Stream Mode

This section presents the comparison of the two streamed data standards used to convey IP video data: the DMB mechanism MPEG2-TS over MSC Stream Mode; and the BT Movio mechanism IP/MSC Datagroups/MSC EPM. Figure 4.47 shows the different performance of the top level PDUs in these standards. It is apparent that the MPEG2-TS/MSC Stream Mode has a more favourable channel performance than IP/MSC Datagroups/MSC EPM. This can be attributed to the greater amount of overhead introduced by the MSC SPM packets, and the additional header information necessary to carry the RS packets for the MSC EPM FEC frame.

The channel performance of a 1,000,000 byte object over the same channels can be seen in figure 4.48, and confirms the conclusion that this model leads us to: IP/MSC Datagroups/MSC EPM performs less well than MPEG2-TS/MSC Stream Mode.

It is evident that the models can find no difference between the FEC protection provided by the two standards, and that both models state that the error protection is ample to cope with a channel affected by a Uniform Random BER.

Comparing MPEG2-TS over MSC Stream Mode with IP over MPE-FEC over DVB-h

MPEG2-TS over MSC Stream Mode was chosen by the DMB community to be an alternative to video data over DVB mechanisms. Here we compare the closest equivalent in the DVB family of protocols: IP over MPE-FEC over DVB-h. Figure 4.49 shows the comparison of these two standards.

The channel performance of sending one IP packet over an MPE-FEC channel is much lower than the channel performance of sending one PDU over MPEG2-TS/MSC Stream Mode. This is attributed to this model having to send one entire
Figure 4.47: Comparison of the performance of a single PDU carried over IP/MSC Datagroups/MSC EPM with a single PDU carried over MPEG2-TS/MSC Stream Mode
Figure 4.48: Comparison of the performance of IP/MSC Datagroups/MSC EPM with MPEG2-TS/MSC Stream Mode
Figure 4.49: Comparison of the performance of a single PDU sent over MSC Datagroups/MSC EPM with a single IP packet sent over MPE-FEC/DVB-h

MPEG2-TS/MSC Stream Mode and MPE-FEC over DVB-h Derived From The Mathematical Model

Comparison Of MPEG2-TS/MSC Stream Mode and MPE-FEC over DVB-h Derived From The Mathematical Model

Figure 4.49: Comparison of the performance of a single PDU sent over MSC Datagroups/MSC EPM with a single IP packet sent over MPE-FEC/DVB-h

MPE-FEC frame, with the part of the MPE-FEC frame not filled with the IP packet, or the RS information, being filled with padding. Figure 4.50 displays the same channel, but carrying a 1,000,000 byte object. Here the performance of IP/MPE-FEC is lower than the performance of MPEG2-TS/MSC Stream Mode, and this can be attributed to the additional RS FEC data that is required to be sent by MPE-FEC in this model. It is worth remembering that this model applies the most pessimistic configuration, where we assume that no puncturing of the RS data is made.

Comparing IP over datagroups over MSC EPM with IP over MPE-FEC over DVB-h

Figure 4.51 shows the comparison of carousel IP data over datagroups over MSC EPM and MPE-FEC over DVB-h.

It can be seen from figure 4.51 that IP over MPE-FEC over DVB-h performs less well than IP over datagroups over MSC EPM using MSC SPM packets greater than 24 bytes in length. The level of overhead required to transmit this MSC EPM channel using 24 byte MSC SPM packets is similar to the MPE-FEC mechanism. Anecdotally, the MPE-FEC mechanism will perform better than MSC EPM as the level of overhead in the MPE-FEC mechanism is due to additional RS data.
Figure 4.50: Comparison of the performance of a 1,000,000 byte object sent IP/MSC Datagroups/MSC EPM with a 1,000,000 byte object sent over IP/MPE-FEC/DVB-h

In this model, however, there is no difference as both RS mechanisms are strong enough to always provide a greater than 95% likelihood of total data receipt on a channel affected by a Uniform Random BER of $10^{-4}$. 
Figure 4.51: Comparison of the performance of IP over datagroups over MSC EPM with IP over MPE-FEC over DVB-h
4.2.11 IP/MSC Datagroups/MSC SPM for Cached Data

In this section the channel performance of cached data sent over IP/MSC Datagroups/MSC SPM, is presented. This builds on the understanding provided by sections 4.2.3 and 4.2.7.

This research has chosen to use IP for the cached data mechanism as one of the most important aspects of this research is the comparison of the different physical layers: DAB and DVB-h. These two provide very different transmission mechanisms for cached data; to provide a comparison, although both provide mechanisms to transmit IP based cached data. For the purposes of comparison, this research assumes that the cached data may be sent over IP channel only.

Performance of the IP and SPM PDUs

Within the unicast constraints of broadcast networks, IP over datagroups over MSC SPM may carry anything that an IP packet may, so we first understand the performance at the network layer. It can be seen when comparing figures 4.14 and 4.52 that IP over datagroups over MSC SPM performs slightly less well than datagroups over MSC SPM in all cases, with the greatest difference observed when the IP payload size is smallest. It can be seen in figures 4.53 and 4.54 that the
Figure 4.53: Performance of the entire range of valid IP packets, given a channel affected by a uniform random BER of $10^{-4}$

same artefacts are present when using IP over datagroups over MSC SPM, as with datagroups over MSC SPM.

**Performance of Objects over IP/MSC Datagroups/MSC SPM**

The channel performance of a cached 1,000,000 byte object sent over a reliable IP/MSC Datagroups/MSC SPM channel is presented in figure 4.55. From this result the understanding that the best configuration for the channel is to use both large MSC Datagroups and large MSC SPM packets. Figure 4.56 presents the same channel, but this time affected with a Uniform Random BER of $10^{-4}$. The understanding that can be taken from the results presented in figure 4.56 is that additional rotations of the carousel increases the likelihood of data arrival, at the cost of channel performance, and that 20 cycles of the carousel is not enough to correct the transmission for large MSC Datagroups. The best configuration can now be assumed to be small MSC Datagroups and large MSC SPM packets.

The reasons for using IP as opposed to the native DAB mechanisms are not due to the performance of the channel, as IP over datagroups over MSC SPM performs worse where the datagroup payload sizes are the same. The reason is that the infrastructure for communicating content from the source to the transmitter uses IP based networks. It is therefore simplest for the broadcaster to not have
Figure 4.54: Improved view of the performance of the entire range of valid IP packets, given a channel affected by a uniform random BER of $10^{-4}$

to reconstruct a DAB transmission, but use the existing packet structure directly, providing the greatest control at the content source.
Figure 4.55: Performance of IP over datagroups over MSC SPM of a 1,000,000 byte object, given a Reliable Channel

Figure 4.56: Performance of IP over datagroups over MSC SPM of a 1,000,000 byte object, given a $10^{-4}$ Uniform Random BER
4.2.12 IP/MSC Datagroups/MSC EPM for Cached Data

In this section the channel performance of cached data sent over IP/MSC Datagroups/MSC EPM, is presented. This builds on the understanding provided by sections 4.2.4 and 4.2.6.

The performance of the PDUs over an IP/MSC Datagroups/MSC EPM channel are the same as have been seen in section 4.2.6, as the RS FEC mechanism is strong enough not to require the data to be repeated to increase the likelihood that it will all arrive correctly. We will now look at the performance of objects over this channel.

Performance of Objects over IP/MSC Datagroups/MSC EPM

The performance of a cache-able 1,000,000 byte object sent over an IP/MSC Datagroups/MSC EPM channel can be seen in figure 4.57. The conclusion that the best configuration for this channel is to use large MSC Datagroups and large MSC SPM packets may be drawn from the results presented in figure 4.57.

Figure 4.57: Performance of IP over datagroups over MSC EPM of a 1,000,000 byte object, given a Reliable Channel

Using the results presented in figure 4.58, the conclusion may be drawn that
the best configuration for this channel is to use large MSC Datagroups and large MSC SPM packets, especially where the size of IP packet and MSC Datagroup fit integrally with each other and with the MSC SPM packets.

4.2.13 MPE-FEC DVB-h

There are a number of standards that can be used to convey cached data over the MPE mechanisms within DVB. Part of what this research is interested in is the comparison of DAB and DVB data channels. In a similar manner to the mechanisms researched in section 4.2.11, this section will look at IP over MPE-FEC over a DVB-h channel, as this is the most equivalent protocol to be used in a comparison between such different physical layers.

The performance of a single PDU sent over MPE-FEC was seen in section 4.2.9. In that section the conclusion was drawn that the error protection of the RS FEC mechanism within MPE-FEC was strong enough to assume that a single PDU would almost certainly arrive correctly after one transmission. We will not re-address this here beyond stating that this is still true, and therefore there is little likelihood of requiring a retransmission of a single PDU.

This section now looks at the performance of a cache-able object sent over such
a channel.

**Performance of Cached Objects over IP over MPE-FEC over DVB-h**

Figure 4.59 depicts the performance of a 1,000,000 byte object carried over an IP over MPE-FEC DVB-h channel. The trend presented in figure 4.59 is for big IP packets and large number of MPE frame rows. The effect of carrying data over the worst-case BER channel is presented in figure 4.60. The results presented in figure 4.60 are the same as those previously seen in figure 4.59, so that we can conclude that the trend is that big IP packets and few MPE rows are best.
Figure 4.60: Performance of one 1,000,000 byte object over IP over MPE-FEC DVB-h, given a channel affected by a uniform random BER of $10^{-4}$.
4.2.14 Comparing the Standards

In this section, we have used the mathematical model to provide an understanding of the performance of various standards over MSC SPM, MSC EPM, and DVB-h. To understand how well a particular standard performs, here the following direct comparisons are made:

- IP over datagroups over MSC SPM with datagroups over MSC SPM
- IP over datagroups over MSC EPM with datagroups over MSC EPM
- IP over datagroups over MSC SPM with IP over datagroups over MSC EPM
- IP over datagroups over MSC EPM with IP over MPE-FEC over DVB-h

Comparing IP over datagroups over MSC SPM with datagroups over MSC SPM

Figure 4.61 shows the performance of a datagroup over MSC SPM channel against the performance of an IP over datagroups over MSC SPM performance. Table 4.11
Table 4.11: Statistical analysis of the effect of IP on carousel transmissions

<table>
<thead>
<tr>
<th>SPM packet size</th>
<th>Difference in Performance</th>
<th>Minimum</th>
<th>Mean Absolute</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>0.07</td>
<td>0.84</td>
<td>6.56</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>0.73</td>
<td>7.78</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>0.64</td>
<td>9.07</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>0</td>
<td>0.60</td>
<td>10.71</td>
<td></td>
</tr>
</tbody>
</table>

shows the statistical difference between datagroups over MSC SPM channel and IP over datagroups over MSC SPM.

From this it can be seen that there are situations where there is a large (10 percent) performance drop when IP is used in addition to datagroups over MSC SPM. However, it can also be seen that the mean absolute difference is small, under 1 percent in all cases.

When the results presented in figure 4.70 and table 4.12, it can be seen that the larger differences in performance are due to the displacement of the sawtooth formation. This is caused by a difference in quantity of overhead, causing the best fit of different PDU's to change.

Comparing IP over datagroups over MSC EPM with datagroups over MSC EPM

Figure 4.62 shows the performance of a datagroups over MSC EPM channel against the performance of an IP over datagroups over MSC EPM performance. Table 4.12 shows the statistical difference between datagroups over MSC EPM channel and IP over datagroups over MSC EPM. From this it can be seen that there are

<table>
<thead>
<tr>
<th>SPM packet size</th>
<th>Difference in Performance</th>
<th>Minimum</th>
<th>Mean Absolute</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>0.13</td>
<td>0.82</td>
<td>25.62</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>0.87</td>
<td>30.42</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>0.84</td>
<td>35.49</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>0</td>
<td>0.81</td>
<td>38.02</td>
<td></td>
</tr>
</tbody>
</table>

situation where there is a large (38 percent) performance drop when IP is used in addition to datagroups over MSC EPM. However, it can also be seen that the mean absolute difference is small, under 1 percent in all cases.

When the results presented in figure 4.70 and table 4.12, it can be seen that the larger differences in performance are due to the displacement of the sawtooth formation.
Figure 4.62: Comparison of the performance of IP over datagroups over MSC EPM and datagroups over MSC EPM.
formation. This is caused by a difference in quantity of overhead, causing the best fit of different PDU s to change.

**Comparing IP over datagroups over MSC SPM with IP over datagroups over MSC EPM**

Figure 4.63 shows the comparison of the performance of IP over datagroups over MSC SPM and IP over datagroups over MSC SPM EPM. It can be seen in figure 4.63 that the best performance for the two standards are at opposite ends of the x-axis.

With the understanding that this mathematical model has provided, it appears that there are consequences when configuring these protocols:

- If the most optimum configuration for MSC SPM is used, MSC EPM performs badly.

- If the most optimum configuration for MSC EPM is used, MSC SPM has a negligible chance of arrival within 20 rotations of the carousel.

This finding implies that, although on paper MSC EPM is backwardly compatible with MSC SPM, in practice the effect on performance means that MSC EPM will not gracefully degrade for MSC SPM only receivers.
Figure 4.63: Comparison of the performance of IP over datagroups over MSC SPM and IP over datagroups over MSC EPM
Comparing IP over datagroups over MSC EPM with IP over MPE-FEC over DVB-h

Figure 4.64 shows the comparison of carousel IP data over MSC Datagroups over MSC EPM and MPE-FEC over DVB-h. It can be seen from figure 4.64 that IP over MPE-FEC over DVB-h performs less well than IP over datagroups over MSC EPM using MSC SPM packets greater than 24 bytes in length. The level of overhead required to transmit this MSC EPM channel using 24 byte MSC SPM packets is similar to the MPE-FEC mechanism. Anecdotally, the MPE-FEC mechanism will perform better than MSC EPM as the level of overhead in the MPE-FEC mechanism is due to additional RS data. In this model, however, there is no difference as both RS mechanisms are strong enough to always provide a greater than 95% likelihood of total data receipt on a channel affected by a Uniform Random BER of $10^{-4}$.
4.3 Enhanced Mathematical Model

Section 4.2 presented the results of the simple mathematical models. Two types of generic mathematical models were presented in chapter 3, the key difference being that the enhanced model employed iteration across a list of previously determined values of PDU. Although iterating through these values will provide a better model, it is also more time consuming.

There is another mathematical method presented in section 3.8 that models a transmission much closer to a real transmission at the cost of not being generic, and this is presented later. Here we present the differences between the MSC SPM mathematical models and the different models of the FEC mechanisms. The enhanced models presented here are the basis for the mathematical results from the software simulation, which will allow us to verify the results of the simulator.

4.3.1 Comparison of the performance of the MSC Datagroups/MSC SPM models

This section compares the results of the two MSC SPM models, to determine the added value provided by the enhanced model. First we will see the differences when modelling a single MSC Datagroup sent over an MSC SPM channel, and then will go on to review the differences in the models for the transmission of objects.

Comparison of the performance of PDUs using the Simple Model and the Enhanced Model

Figure 4.65 shows the difference between the simple and enhanced mathematical models for datagroups over MSC SPM over a reliable channel. Table 4.13 shows the statistical differences between the two models. It can be concluded from

<table>
<thead>
<tr>
<th>Metric</th>
<th>24</th>
<th>48</th>
<th>72</th>
<th>96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Difference (%)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum Difference (%)</td>
<td>0.00</td>
<td>17.71</td>
<td>26.04</td>
<td>33.33</td>
</tr>
<tr>
<td>Mean Absolute Difference (%)</td>
<td>0.00</td>
<td>0.47</td>
<td>1.02</td>
<td>1.57</td>
</tr>
</tbody>
</table>

Table 4.13: Statistical differences between the Simple and Enhanced Models for one datagroup over MSC SPM over a reliable channel

table 4.13 that there are some cases where the models are identical. It can also be seen that there are cases where they differ. When these results are compared to figure 4.65, it can be seen that these are all rounding errors. The simple model
Figure 4.65: Comparison of the Simple Model with the Enhanced Model for one datagroup over MSC SPM over a reliable channel

(a) 24 Byte SPM packets
(b) 48 Byte SPM Packets
(c) 72 Byte SPM Packets
(d) 96 Byte SPM Packets
does not perform as well as the enhanced model where an MSC datagroup does not fit integrally into a number of MSC SPM packets.

The 24 byte MSC SPM case is identical in the two models. The simple model is able to derive the correct number of MSC SPM packets for the datagroup as this is easy in the case where there is only one datagroup. The other cases do not match perfectly as the simple model is unable to select a different MSC SPM packet to convey a small amount of data, which would fit into a smaller MSC SPM packet. The best performance case is always identical as the most efficient transmission is where the data fits integrally into a number of the largest MSC SPM packets. This is correctly modelled by both the simple and enhanced models.

Both models give a similar impression of the performance of the channel, with the simple model providing a pessimistic view of the worst case.

Figure 4.66 shows the same comparison already made in this section, as was seen in figure 4.65, but with the impact of a Uniform Random BER of $10^{-4}$, the worst case for a DAB channel. It can be seen in figure 4.66 that the performance

![Figure 4.66: Comparison of the Simple Model with the Enhanced Model for one datagroup over MSC SPM over a channel affected by a Uniform Random BER of $10^{-4}$](image)

is reduced when compared to the case of the reliable channel, which we saw in section 4.2.3. The difference between the two channels is small, with the largest
difference being where the two models determine when the receiver has a smaller than 95% probability of receiving all of the data. This can be seen in table 4.14.

<table>
<thead>
<tr>
<th>Metric</th>
<th>SPM Packets (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Difference</td>
<td>0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>Maximum Difference</td>
<td>0.00 16.67 44.44 50.00</td>
</tr>
<tr>
<td>Mean Absolute Difference</td>
<td>0.00 0.68 1.45 1.95</td>
</tr>
</tbody>
</table>

Table 4.14: Statistical differences between the Simple and Enhanced Models for one datagroup over MSC SPM over a reliable channel.
Comparison of the performance of Objects using the Simple Model and the Enhanced Model

Figure 4.67 shows the difference between the simple and enhanced mathematical models for one 1,000,000 byte object over datagroups over MSC SPM over a reliable channel. Table 4.15 shows the statistical differences between the two models.

It can be seen from table 4.15 that there are some cases where the models are identical. It can also be seen that there are cases where they differ. When these results are compared to figure 4.67, it can be seen that these are all rounding errors, for the same reasons presented in section 4.3.1.

<table>
<thead>
<tr>
<th>Metric</th>
<th>SPM Packets (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Difference</td>
<td>0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>Maximum Difference</td>
<td>0.63 17.71 26.04 33.33</td>
</tr>
<tr>
<td>Mean Absolute Difference</td>
<td>0.16 0.65 1.21 1.76</td>
</tr>
</tbody>
</table>

Table 4.15: Statistical differences between the Simple and Enhanced Models for one 1,000,000 byte object over datagroups over MSC SPM over a reliable channel.
When tables 4.13 and 4.15 are compared, it can be seen that there is a difference in the 24 byte MSC SPM models in table 4.15 which was not present in table 4.13. In the one datagroup case, the simple model correctly found the number of MSC SPM packets required, as there was only ever one datagroup transmitted. The enhanced model also found the correct number of packets. This lead to there being no difference between the two models. In this one 1,000,000 byte case, the simple model did not always derive the correct number of MSC SPM packets, as there were more than one datagroup. This lead to the small discrepancy between the two models.

Both models give a similar impression of the performance of the channel, with the simple model providing a pessimistic view of the worst case.

Figure 4.68 shows the same comparison already made in this section, as was seen in figure 4.67, but with the impact of a Uniform Random BER of $10^{-4}$, the worst case for a DAB channel. A clear explanation of a carousel artefact can be seen in figure 4.68c where the performance of the simple model reduces by an increase from 8 to 9 carousel rotations at a MSC Datagroup payload size of 258 bytes, to be later increased by a reduced number of carousel rotations from 9.
to 8 at a MSC Datagroup payload size of 300 bytes. The enhanced model, which is more adept at correctly discovering the MSC SPM packet size is not affected by this artefact, as the total length of MSC SPM packets being sent is the minimum required.

It can be seen in figure 4.68 that the performance is reduced when compared to the case of the reliable channel, which we saw in section 4.2.3. The difference between the two channels is small, with the largest difference being where the two models determine when the receiver has a smaller than 95% probability of receiving all of the data. This can be seen in table 4.16.

<table>
<thead>
<tr>
<th>Metric</th>
<th>SPM Packets (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Minimum Difference</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum Difference</td>
<td>0.00</td>
</tr>
<tr>
<td>Mean Absolute Difference</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4.16: Statistical differences between the Simple and Enhanced Models for one datagroup over MSC SPM over a reliable channel

**4.3.2 Comparison of the models of the RS FEC mechanisms**

Both MSC EPM and MPE-FEC channels have two concurrent error protection mechanisms: CRCs and RS FEC. They both have two conceptual tables: the application data table and the RS data table. The application data table of both MSC EPM and MPE-FEC have a PDUs which have a CRC checksum independent of the RS mechanism carried in the RS data table.

If the application data table arrives correctly at the receiver, but the channel is affected by a bursty error profile which breaks the RS mechanism, both standards recommend that the receiver ignore the RS data, and rely on the CRC checksums.

The models for the case where all of the application data table arrives correctly and the RS data table didn’t or that the whole frame arrived in a state where the RS mechanism could correct any errors in the frame was presented in sections 3.4 and 3.7.

The results of the comparison of the two models, for both standards, do not provide enough of a difference to be noticed above the computer’s rounding mechanism when using modelling a channel affected by a Uniform Random BER of $10^{-4}$.  

155
4.4 Software Simulation

In this section the simulator's output is compared with expected results from the mathematical model. Once this comparison has been made, further simulations will then be conducted.

There were four All Zeros channels that were provided by Arqiva for real performance analysis. The channel configurations are outlined in table 3.3. As the channels have a fixed bandwidth, the interesting configurations of the simulator are the ones that are similar to these. Therefore the experiments shall concentrate on these four bandwidths.

From the results that have been seen already in this chapter, the difference between IP over datagroups over MSC SPM and datagroups over MSC SPM, and IP over datagroups over MSC EPM and datagroups over MSC EPM are slight. The broadcasters have expressed a strong preference to using IP, as this makes communication between content source and transmitter easier, and the hardware in transmitters and receivers more standard and cheaper therefore.

The reception quality, and therefore the channel performance, in this section is calculated differently from the mathematical models which we have used thus far in this research. The reception quality is calculated as the mean of the quantity of data received correctly for all iterations of the experiment. This means that we no longer have to assume that all of the data is received with at least a 95% certainty of correct arrival. In the case of the streamed reception, the quantity of data received in that transmission is presented. In the case of carousel data, the data is repeated until all of the data is received, or 20 rotations of the carousel are transmitted.

There are six All Zero Cambridge profiles used in this section, all captured in the manner described in section 3.10. The All Zeros 1 and All Zeros 4 channels arrived in one capture process. The All Zeros 2 and All Zeros 3 channels are captured in two halves as the receiver broke in reception, which required the capture process to be restarted, therefore that are two capture profiles each for All Zeros 2 and All Zeros 3.

Each data point in the simulator represents a configuration that has been iterated many times. In both the Uniform Random BER generated error file, and All Zeros capture files, the simulator repeats each iteration starting a byte after the previous iteration finished, until there is no more file. A direct comparison of the configurations may be made because of this. However, it should be noted that the All Zeros experiments have had a different number of iterations performed on them as the different configurations have required a different number of bytes.

This section now presents the validity of the simulator, and then continues by
simulating the standards to provide an understanding of the best configuration for each standard.

4.4.1 Validating the Simulator

The mathematical model has been used in this research to show the probability of receipt of all of the data transmitted. The simulation will produce data in addition to the probability that all of the data arrived. To verify that the simulator is performing in the correct manner, here we compare the expected results from mathematical model with the observed results from the simulation.

For this comparison two mechanisms were chosen: streamed data over MSC Datagroups/MSC SPM; and cached data over MSC Datagroups/MSC SPM. These were chosen as the simple model, enhanced model and simulation all have a very similar set of constraints, thus providing the fairest comparison.

Stream Transmissions

The expected results, for a stream transmission, derived from the mathematical model and the observed simulation results, can been seen in figure 4.69. Comparing the observed and the expected values will show how similar the two sets of data are. Table 4.17 lists the statistical similarities. It can be observed that the expected results are consistently lower than the observed results. This is due to rounding error with small probabilities.

From the analysis provided in figure 4.69 and table 4.17 we may conclude that the simulation is a valid interpretation of the stream data over MSC Datagroups/MSC SPM broadcasting standard.

Carousel Transmissions

The expected results, for a carousel transmission, derived from the mathematical model and the observed simulation results, can been seen in figure 4.70. Comparing the observed and the expected values shows how similar the two sets of data are. Table 4.18 lists the statistical similarities.

From the analysis provided in figure 4.70 and table 4.18 we may conclude that the simulation is a valid interpretation of the carousel data over MSC Datagroups/MSC SPM broadcasting standard.
Figure 4.69: Comparison of observed simulation results with expected enhanced model results for streamed data sent over MSC Datagroups/MSC SPM

<table>
<thead>
<tr>
<th>Metric</th>
<th>SPM Packets (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Minimum Difference (%)</td>
<td>2.74e-3</td>
</tr>
<tr>
<td>Maximum Difference (%)</td>
<td>9</td>
</tr>
<tr>
<td>Mean Absolute Difference (%)</td>
<td>1.9</td>
</tr>
</tbody>
</table>

Table 4.17: Statistical analysis of the comparison of observed simulation results with expected enhanced model results for streamed data sent over MSC Datagroups/MSC SPM
Figure 4.70: Comparison of observed simulation results with expected enhanced model results for carousel data sent over MSC Datagroups/MSC SPM

<table>
<thead>
<tr>
<th>SPM Packets (Bytes)</th>
<th>Metric</th>
<th>24</th>
<th>48</th>
<th>72</th>
<th>96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Difference (%)</td>
<td>6.56e-3</td>
<td>1.13e-2</td>
<td>1.13e-2</td>
<td>1.13e-2</td>
<td></td>
</tr>
<tr>
<td>Maximum Difference (%)</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Mean Absolute Difference (%)</td>
<td>2.4</td>
<td>2.6</td>
<td>2.7</td>
<td>2.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.18: Statistical analysis of the comparison of observed simulation results with expected enhanced model results for carousel data sent over MSC Datagroups/MSC SPM
4.4.2 Streamed data over IP over MSC Datagroups over MSC SPM

This section looks at the performance of IP/MSC Datagroups/MSC SPM using the simulator. The results here can be compared to the results of the model of this channel in section 4.2.1. A significant difference in the results is apparent when such a comparison is made: the simulation results provide a different understanding of reception quality, and therefore channel performance. The reception quality in the simulation is defined as the quantity of data which arrived correctly, against the quantity that was sent. In the model we assumed that all of the data arrived with a probability of at least 95%, the simulation can state how much of the data was received, and the mean of this value across many iterations of the same experiment is then used to provide the average reception quality. This provides a smoother channel performance metric that degrades with the impact of error more gracefully than the mathematical models.

Performance of IP and MSC SPM PDUs

A change in bandwidth does not only effect the speed of delivery but also affects the size of the logical frame, which directly affects the size of the SPM packets.

The performance of IP over datagroups over MSC SPM 8, 96, 128 and 384kbps channels can be seen in figure 4.71. It can be seen in figure 4.71a that all four lines perform in the same manner. This is because on a 8kbps channel the maximum SPM packet size possible is 24 bytes. If a maximum possible SPM payload size is specified as 48, 72 or 96 bytes, it reduces to the maximum that the logical frame can carry. In an 8kbps channel they all reduce to a 24 byte SPM packet. The results presented in figure 4.71 that the channel performance degrades more gracefully than as modelled previously in figure 4.7. This is due to the ability of the simulator to average the quantity of correctly received data, rather than make the assumption required to build the model, which states that all or nothing is correctly received.

There is a strong similarity to the trend of the channel performance presented in the simulation results to the results provided by the model: they both recommend a small IP packet size to get the best channel performance. The simulation and the model differ on the optimum size of MSC SPM packets. The simulation shows the best performance provided by the 96 byte packet, where the model deemed the 48 byte packet the optimum, as previously seen in section 4.2.1. This is attributed to the more forgiving reception quality calculation within the simulation, rather than an increased likelihood of all of the data arriving.
Figure 4.71: Channel performance of a single IP packet over MSC Datagroups over MSC SPM over a channel affected by a Uniform Random BER of $10^{-4}$
Performance of Objects over IP/MSC Datagroups/MSC SPM

Having now seen the performance of a single IP packet sent over an MSC Datagroups/MSC SPM channel, the channel performance of a 1,000,000 byte object is presented. Figure 4.72 shows the channel performance for the four bandwidths, all affected by a Uniform Random BER of $10^{-4}$.

![Graphs showing channel performance](image)

Figure 4.72: Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC SPM over a channel affected by a Uniform Random BER of $10^{-4}$.

The shape of the results presented in figure 4.72 indicate a strong similarity to the results of the single IP packet sent over the same channels, seen in figure 4.7, although the channel performance is reduced in all cases for the 1,000,000 byte object. This reduction indicates the inevitable increase of loss due to sending a larger quantity of data over a lossy channel. Figure 4.73 now shows the same channel, but sent over the Cambridge All Zero channels.

The results presented in figure 4.73 show a number of interesting things. It is clear from the shape of the graph that the Cambridge All Zero channel profiles do not conform to a Uniform Random BER error profile, which is due to the bursty nature of the Cambridge All Zero Channels.

To a greater or lesser extent, all the Cambridge All Zero channel profiles show an improvement with increasing PDU payload size, followed by a degradation of
channel performance. This is akin to the Uniform Random BER case, although with a less aggressive degradation than seen with the Uniform Random BER error profile.

The All Zeros 1, 2 part 2 and 3 part 2 all are affected by a lot of errors. All Zeros 2 part 2 and 3 part 2 were received in the second half of the rotation around Cambridge. This was further from the transmission tower so had a lower general signal strength.

The All Zeros 2 part 1 and 3 part 1 were both received closest to the transmission tower. There were features, including bridges and traffic, that affected the

Figure 4.73: Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC SPM over the Cambridge all-zero channels

163
signal strength, but this induced fewer errors than the second parts as the general signal strength was higher.

The All Zeros 4 channel is relatively unaffected by errors.
4.4.3 Streamed data over IP over MSC Datagroups over MSC EPM

The performance of streamed data over IP over MSC Datagroups over MSC EPM as this is the mechanism that was used in the BT Movio experiments, designed as an alternative to the DMB mechanism of MPEG2-TS over MSC Stream Mode. Figure 4.74 presents the performance of the PDUs for streamed data over IP over MSC Datagroups over MSC EPM.

This section looks at the channel performance of streamed data sent over an IP/MSC Datagroups/MSC EPM channel, as presented by the simulation. First the channel performance of a single IP packet sent over MSC Datagroups/MSC EPM is presented, followed by the performance of objects sent over the same channel.

Performance of IP and MSC SPM PDUs

Using the results from section 4.4.2, the best channel performance is provided when the largest possible MSC SPM packets are used. In this section, the results of 96 byte MSC SPM packets are presented, with the understanding that the simulation will take into account the size of the logical frame and the object size, and the largest possible, and the smallest needed, MSC SPM packet will be used.

The results of the channel performance of one IP packet sent over MSC Datagroups/MSC EPM over the Cambridge All Zero channels can be seen in figure 4.74.

A number of things can be seen in figure 4.74:

- The channel performance of the All Zeros 1 channel has a reduced performance, which is attributed to the largest MSC SPM packet possible in that channel being a 24 byte packet, which increases the overhead, thus reducing the transmission efficiency.

- The channel performance of all of the All Zeros channels degrades slightly with larger IP packet sizes. This is attributed to losing a greater quantity of data when there is a packet loss when more data is sent in one PDU.

- The errors in All Zeros channel 2 part 2 and All Zeros channel 3 part 2 are high enough to cause a substantial drop in channel performance.

It can be taken from the results presented in figure 4.74 that the most optimum transmission configuration is when the data fits integrally into one MSC EPM FEC frame.
The performance of a 1,000,000 byte object sent over IP/MSC Datagroups/MSC EPM is presented in figure 4.75. The results presented in figure 4.75 are similar to those presented in figure 4.34, demonstrating again that the error correction strength of the RS (204,188) mechanism within MSC EPM is strong enough to correct the worst case Uniform Random BER of $10^{-4}$. The results of the same configuration but over the Cambridge All Zero channels is presented in figure 4.76.

The results in presented in figure 4.76 demonstrate that the error correction provided by the RS (204,188) mechanism within MSC EPM is not strong enough to always correct the burst errors found in the Cambridge All Zero channels, although it does correct a substantial number. In the majority of the channels, the channel performance does not degrade with larger IP packet sizes, although the error correction does not provide enough protection against loosing channel performance in the case of the Cambridge All Zeros 3 part 2 channel, seen in figure 4.76e. This channel does noticeable degrade in a similar manner to that found in section 4.4.2.
Figure 4.75: Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over a channel affected by a Uniform Random BER of $10^{-4}$.
Figure 4.76: Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels
4.4.4 Streamed Data over MPEG2-TS over MSC Stream Mode

In this section we see an MPEG2-TS stream over MSC Stream Mode, affected by the errors that were collected in the Cambridge trials. This can be seen in figure 4.77.

![Software Channel Performance of Graph 83](image)

Figure 4.77: The performance of MPEG2-TS over MSC Stream Mode over the channels affected by the Cambridge reception data

It can be seen that the FEC mechanism built into MSC stream mode is strong enough to negate the impact of error in four of the six reception cases, and that in these four cases, our understanding so far that large PDUs perform best is still valid.

In the two cases where the FEC is not strong enough to negate the impact of the errors that have been introduced into the channel present an interesting new piece of understanding. The All Zeros 2 case demonstrates a similar trend to the unprotected MSC SPM mechanism, where small PDUs are best, however the All Zeros 3 case shows that the middle sizes of PDU perform best.

The channel performance of a 1,000,000 byte object sent over a channel using PDUs similar to IP packets is presented in figure 4.78.

Due to the fragmentation of IP packets, the performance presented in figure 4.78 is low, although the channel quality is very high in this result. Figure 4.79
Figure 4.78: The performance of a 1,000,000 byte sent over MPEG2-TS over MSC Stream Mode over the channels affected by the Cambridge reception data shows this object sent over the Cambridge All Zero channels.

The FEC protection provided in MPEG2-TS/MSC Stream Mode is fairly effective in removing the impact of error within the channel, seen as most of the results are very similar.
Figure 4.79: Channel performance of a 1,000,000 byte object over MPEG-2 TS/MSC Stream Mode over the Cambridge all-zero channels
4.4.5 Streamed Data over IP over MPE-FEC

Figure 4.80 presents the performance of a single 1,000,000 byte object conveyed by IP over MPE-FEC over DVB-h over a channel affected by a uniform random BER rate of $10^{-4}$.

![Software Channel Performance of streamed object over IP/MPE-FEC/Uniform Random BER $10^{-4}$](image)

It can be seen in figure 4.80 that the performance of the transmissions all step, which is caused by the reducing need for additional MPE-FEC frames as the IP packet size increases reducing the required overhead for transmission. Each of the configurations achieve different levels of performance due to the different amounts of padding required when a different number of MPE-FEC frames is used.

In practice, DVB-h would never be sent over a DAB channel. For the purposes of this research, it is assumed that the Cambridge All Zero profiles are representative of a DVB-h physical channel, which will allow us to directly compare DVB-h with the native DAB standards. Figure 4.81 shows the performance of these configurations sent over the Cambridge All Zero channels.

The results presented in figure 4.81 show that the DVB-h mechanism is affected by the error bursts found within the Cambridge All Zero profiles, although the shape is similar to the Uniform Random BER case, so it is clear that the FEC mechanism is good at protecting this channel.
Figure 4.81: Channel performance of a 1,000,000 byte object over IP packets over MPE-FEC over the Cambridge all-zero channels
4.4.6 Comparison of Streamed Data over IP/MSC Datagroups/MSC SPM with Streamed Data over IP/MSC Datagroups/MSC EPM

Previously in section 4.2.5 the comparison of streamed data over MSC SPM and MSC EPM as modelled mathematically was presented. Here the results of the simulation of these two standards, IP/MSC Datagroups/MSC SPM and IP/MSC Datagroups/MSC EPM, seen in sections 4.4.2 and 4.4.3 are compared in figures 4.82, 4.83, 4.84, 4.85, 4.86 and 4.87.

Across all of these results, the best configuration is found with payload sizes of less than 2,000 bytes, and typically the best compromise between the MSC SPM and MSC EPM channels is found around 1,000 byte payloads. The All Zero channels 1, 2 and 4 all present that MSC SPM has an advantageous channel performance over MSC EPM. All Zero channel 3 presents that MSC EPM has a stronger channel performance.

The finding that there is a valid compromise between these standards is a different result than that found in the Uniform Random BER case, where the model states that the most optimum solution for these standards are at odds with each other. It is worth noting that the most optimum configuration found here is deemed statistically unlikely to arrive in the Uniform Random case.
Figure 4.82: Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1
Figure 4.83: Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 1
Figure 4.84: Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 2
Figure 4.85: Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 1
Figure 4.86: Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 2
Figure 4.87: Comparison of channel performance of a 1,000,000 byte object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1

(a) 24 Byte MSC SPM Packets
(b) 48 Byte MSC SPM Packets
(c) 72 Byte MSC SPM Packets
(d) 96 Byte MSC SPM Packets
4.4.7 Comparison of Streamed Data over MPEG2-TS/MSC Stream Mode with Streamed Data over IP/MSC Datagroups/MSC EPM

In this section we compare the results that we have already seen in sections 4.4.3 and 4.4.4, to understand the relative performance of these competing streamed broadcast data standards. Figure 4.88 shows the relative performance of a single 1,000,000 byte object sent over channels affected by each of the Cambridge captured error profiles.

Figure 4.88: Channel performance of a 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels
It can be seen in figure 4.88 that MPEG2-TS/MSC Stream Mode has a much more consistent performance than IP/MSC EPM over the same channel.

Although the particular configuration used in the MPEG2-TS/MSC Stream Mode experiment is less efficient than that seen in the MSC EPM one, it is clear that the affect of the convolutional interleaver has a dramatic effect on the FEC capabilities of the RS (204,188) mechanism, as the reception quality is not adversely affected with large PDUs. It is clear from these results that the mechanism for interleaving the RS rows within the MSC EPM FEC frame is not as strong and the burst errors found in the Cambridge All Zero channels are able to break the FEC protection with larger PDUs.
4.4.8 Comparison of Streamed Data over IP/MSC Datagroups/MSC EPM with Streamed Data over IP/MPE-FEC

We have previously seen the channel performance of both IP/MSC Datagroups/MSC EPM and IP/MPE-FEC in sections 4.4.3 and 4.4.5. Figure 4.89 provides a comparison of these two standards.

Figure 4.89: Channel performance of a 1,000,000 byte streamed object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels

As described in chapter 3, the FEC protection in the MSC EPM standard
and the MPE-FEC standard differ from each other: MSC EPM uses RS (204,188) where MPE-FEC is assumed to use RS (255,191). The comparison of these mechanisms shown in figure 4.89 indicates that these two levels of protection provide a similar channel performance. All Zeros 3 part 2 presents the best case for MPE-FEC, where it performs as well as 96 byte MSC SPM packets carried in an MSC EPM channel. In all of the other All Zero channels the MPE-FEC standard performs with a similar channel performance of the 24 byte MSC SPM packets over MSC EPM.
4.4.9 Cached data over IP over MSC Datagroups over MSC SPM

The simulated results of the stream standards have now been compared and presented. This section now presents the results of carousel data sent over IP/MSC Datagroups/MSC SPM. As mentioned at the start of this chapter, streamed data and cached data have different uses and a different value on loss. Streamed data tends to cope well with small amounts of error, where carousels tend to cope badly with loss.

The simulations in this section repeat the carousel twenty times, or until the data arrives. If the entire carousel has not arrived in twenty cycles, the quantity of data that has been correctly received is used in the reception quality metric. Although this is not ideal to have missing data at the receiver, it is normally better to have something, rather than nothing, so it is assumed that whatever the quantity of data, it is useful.

Figure 4.90 presents the simulation results of a single 1,000,000 byte object sent over IP/MSC Datagroups/MSC SPM on a channel affected by a Uniform Random BER of $10^{-4}$. The results presented in figure 4.90 show a similar formation as was
seen in section 4.2.11. It is clear from the shape of the graph that more than 20 carousel rotations are required to receive all of the data when the IP packet size is greater than 1,000 bytes. In the results of the model, the channel performance dropped to 0 at that point; the simulation provides a smoother degradation as the IP payload size increases.

Figure 4.91 shows the same channel configuration, but affected by the Cambridge All Zero channels. Although it is obvious from the results in figure 4.91 that there is an element of luck involved with finding the most optimum channel configuration, there is also a clear trend that the larger IP packet sizes have a
similar channel performance to smaller ones, which is in contrast to the Uniform Random BER channel.
4.4.10 Cached data over IP over MSC Datagroups over MSC EPM

This section presents the channel performance of a carousel object sent over IP/MSC Datagroups/MSC EPM. Figure 4.92 shows the results of a 1,000,000 byte object sent over this channel affected by a Uniform Random BER of $10^{-4}$. It's clear from the results presented in figure 4.92 that the data arrived in one carousel rotation, and therefore the biggest IP packet sizes are the most optimum.

Figure 4.93 now shows the same channel configuration, this time affected by the Cambridge All Zero channels.

Cambridge All Zero channels 1, 2 and 4 all follow the trend of an improved channel performance with greater IP packet sizes. Cambridge All Zeros channel 3 have a peak channel performance around 500 byte IP packet payload sizes.
Figure 4.93: Channel performance of a cached 1,000,000 byte object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels
4.4.11 Cached Data over IP over MPE-FEC

This section presents carousel data over IP/MPE-FEC. Figure 4.94 shows the channel performance of such a channel over a Uniform Random BER of $10^{-4}$. The strength of the FEC mechanism within MPE-FEC is strong enough that the entire 1,000,000 byte object arrives in one carousel rotation. Figure 4.95 shows the same object, sent over the same channel, but affected by the Cambridge All Zero channels.

Cambridge All Zeros 1 does not have enough data in it to complete a DVB-h MPE-FEC carousel transmission, so has not been included.

The other All Zero channels all display a trend similar to that seen with the Uniform Random BER results, although all of the All Zero channels have been affected to different extents with the errors that are present in the channels.
Figure 4.95: Channel performance of a carousel 1,000,000 byte object over IP packets over MPE-FEC over the Cambridge all-zero channels
4.4.12 Comparison of Cached Data over IP/MSC Datagroups/MSC SPM with Cached Data over IP/MSC Datagroups/MSC EPM

This section builds on the understanding provided in sections 4.4.9 and 4.4.10, providing a comparison of carousel data sent over IP/MSC Datagroups over both MSC SPM and MSC EPM.

Figures 4.96, 4.97, 4.98, 4.99, 4.100 and 4.101 compare the channel performance of the two standards sent over the Cambridge All Zero channels.

In all of the comparisons presented, it’s clear that there is an advantage in using the error protection included within MSC EPM, rather than the unprotected MSC SPM standard. The best configuration across all of the graphs is to use IP packet sizes of about 1,000 bytes in length.

Figure 4.96: Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1
Figure 4.97: Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 1
Figure 4.98: Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 2 part 2
Figure 4.99: Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 1
Figure 4.100: Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 3 part 2
Figure 4.101: Comparison of channel performance of a 1,000,000 byte carousel object over IP/MSC Datagroups/MSC SPM with IP/MSC Datagroups/MSC EPM over the Cambridge All Zeros 1
4.4.13 Comparison of Cached Data over IP/MSC Datagroups/MSC SPM with Cached Data over IP/MPE-FEC

This section compares the channel performance of a carousel object sent over IP/MSC Datagroups/MSC SPM with the same object sent over IP/MPE-FEC. Figure 4.102 provides the comparison of these standards over the Cambridge All Zero channels.

Where the channel is not affected heavily by error, it is best to use the MSC EPM mechanism, as channel performance is better in this case. Where the channel is more heavily affected by error, for example in the case of the channel over All Zeros 3 part 1 in figure 4.102c, the stronger MPE-FEC mechanism provides a better channel performance.

It can be taken from these results that the stronger FEC protection in MPE-FEC only affects the results on the worst channels, and that in most cases the additional overhead required to convey the additional FEC data negates the benefit to the channel.

4.5 Summary

Chapter 1 outlined the areas of interest for this research. In chapter 2 provided an understanding of what digital radio standards exist for different applications and provided an in depth understanding of the digital broadcast radio standards DAB, DMB and DVB-h. In chapter 3 models and a simulator for these standards was developed, and a new network metric was defined, called Channel Performance. In this chapter we have seen the results provided by these new models and the simulator, and this has provided new knowledge:

• The simulator matches the models where the configuration of the experiment should allow a match. This provides confidence in the simulator.

• The channel performance metric shows that in some situations the efficiency of the channel is most important, and in others the quality of the reception is more important, whilst ignoring the impact of time. The same standard on different bandwidth links can be directly compared to show the affect of bandwidths on protocols. An example of this can be seen when using a low bandwidth DAB channel, which performs less well due to an increase in required overhead by the DAB protocols.

• On channels with high error rates, smaller PDUs are better as the likelihood of loosing the PDU is smaller.
Figure 4.102: Channel performance of a 1,000,000 byte carousel object over IP packets over MSC Datagroups over MSC EPM over the Cambridge all-zero channels
• On channels with low error rates, larger PDUs are better as there is a lower overhead requirement.

• On the evidence of the all zero channel data, there is limited benefit to including FEC techniques for streamed applications. This is because the increase of overhead isn't significantly different from the gain in reliability.

• On the evidence of the all zero channel data, there is a significant benefit in including FEC techniques for cached data applications. This is different from the streamed data model and is because the overhead of an additional carousel rotation is substantially larger than the increase in overhead of the FEC data.

• Strong convolutional interleavers have a significant benefit to FEC techniques, as shown using the all zero channel data. This is because the FEC techniques work best when the error profile is Uniform Random; errors from external factors tend to be bursty in nature, and the convolutional interleaving is good at dispersing the errors through the channel in what appears to be a more Uniform Random pattern.

We now go onto drawn the conclusions from this work in chapter 5.
Chapter 5

Conclusions

Chapter 1 outlined the fields of interests for this research. The digital broadcast mechanisms DAB, DMB and DVB-h where outlined in chapter 2. There was a description of both a mathematical model and a broadcast digital radio simulator in chapter 3. Chapter 4 provided results from both the mathematical model and the simulator. Now the conclusions of this research are drawn in this chapter.

5.1 The Contributions Of This Thesis

This research has contributed a number of new pieces of knowledge, which include:

- The Channel Performance metric, which is proficient at comparing unlike network configurations and providing a solid understanding of network goodness
- Mathematical models of some existing broadcast digital radio standards
- A software digital broadcast radio simulator, capable of simulating those standards
- A clear understanding of the performance of those broadcast digital radio standards, leading to knowledge of best practice when configuring such networks

5.1.1 Channel Performance

There have always been many different metrics for assessing the performance of a network: efficiency, throughput and delay are viewed as the norm. However, in broadcast networks, especially over lossy channels, neither the efficiency or throughput metrics provide a strong enough understanding of the end-user’s experience. As part of this research a new metric was proposed, shown in equation 3.3,
which incorporated both efficiency and also how much of the transmission was received. This new metric, called channel performance, provides a clearer understanding of lossy digital broadcast channels similar to the goodput metric. Channel performance additionally provides a metric which can be used to compare two channels with different bandwidths, removing the effect of time.

5.1.2 Mathematical Modelling

Often the quickest way to understand a problem is to have a formula with which adjustments to parameters may be made. Although models tend to be simplistic in their approach, this can be invaluable to gaining an understanding of the problem. The models that have been presented in this research provide a good understanding of best practice for setting up a broadcast digital radio transmission mechanism.

5.1.3 Software Simulation

It is sometimes necessary to test a theory for which a model would be too difficult, or time consuming. In these cases a simulator may be a better choice. As part of this research a simulator capable of representing any digital broadcast radio standard, existing, proposed or theoretical, was developed.

This simulator can produce a variety of results including the mean channel performance across a number of iterations of an experiment, providing a better knowledge of channel performance, especially in the case of streamed data.

The results provided by this simulator have provided a clear best configuration of complicated transmission scenarios.

5.1.4 About the standards

There are many digital broadcast radio standards for the broadcast community to choose from. There are many options that the broadcast engineers have when configuring the chosen standard. This work has shown that:

- MSC EPM may gracefully degrade for MSC SPM receivers
- The DAB family can choose to use error correction techniques when transmitting streamed broadcast data
- The DAB family ought to use error correction techniques when transmitting carousel broadcast data
- The DAB family has a better reception quality when using MSC Stream Mode over MSC EPM, as the Forney approach convolutional interleaver provides an error profile better suited to the RS FEC mechanism
Increasing the FEC protection on a channel does not always increase the channel performance

MSC EPM degradation to MSC SPM

A major design aim for MSC EPM was to be fully backward compatible with MSC SPM receivers. The optimum settings provided by the understanding gleaned from the model for optimal performance for the two standards are in opposition, as was seen in figures 4.30 and 4.31, and this was confirmed by the simulation of the Uniform Random BER channel. However, the best compromise according to the simulation using the Cambridge All Zero channels is between 500 and 2,000 bytes, with a common optimum of 1,000 bytes across all of the Cambridge All Zero channels, as seen in sections 4.4.6 and 4.4.12. With the normal MTU over Ethernet networks being 1,500 bytes, it seems reasonable to expect that IP based protocols sent over broadcast data services within DAB to use 1,500 bytes, or even 1,000 bytes, as the standard maximum PDU size, thus removing the need for Jumbo-frame Ethernet capable hardware.

In the case of streamed data, the channel performance tends to be best when using the MSC SPM based protocols, where MSC EPM based protocols tend to do better in the carousel applications. In practice this decision is likely to be both political and financial, both of which are beyond the scope of this research.

On the use of Forward Error Correction

The decision to use error correction will depend on the application sent over the data service. If the data is streamed, it is likely to be tolerant of some error; if the data is cached, it is unlikely to be tolerant of any error.

The use of a FEC mechanism will always reduce the transmission efficiency, and may increase the reception quality, depending on how errant the channel is. It is also clear from this research that the choice to use a FEC mechanism depends on the application.

For a streamed application, the addition of a FEC mechanism does not necessarily increase the channel performance, as seen in section 4.4.6. However, the inclusion of an strong convolutional interleaver, like the Ramsey Approach, with a FEC technique provides a much stronger resilience to error, which can be taken from in section 4.4.7.

Carousel transmissions over a lossy channel do benefit from a FEC technique, as was seen in section 4.4.12.

We have concluded that including or increasing the strength of the FEC mechanism does not always increase the channel performance when analysing real error
profiles, which can be seen in sections 4.4.8 and 4.4.13.

Digital broadcasts should always use error correction with interleaving. When the Eureka 147 project started out it was decided to provide a digital bitstream resilient to error enough for an audio codec to cope with the worst case error profile. Data-casting was not considered as important in the initial concept.

As an observation for any future digital broadcast radio standard, and the tuning of existing standards: it is likely that data, in some form, will be required to be sent over the channel. Therefore the standard ought to be designed with a set of flexible error protection and correction techniques built into the standard from the start. It is reasonable to have many (more than one) levels of error protection.

All of the FEC mechanisms would benefit from the inclusion of a convolutional interleaver to disperse the errors across the FEC mechanism, increasing the chances of error correction. It should be noted that the interleaving would be detrimental to the non-FEC protected mechanisms, as it would increase the likelihood that more PDUs would arrive affected by errors.

5.1.5 On the Impact of Errors

We have seen that the impact of errors can be catastrophic to the quality of reception. The different standards all possess some level of mechanism to deal with error:

**Streamed data over MSC SPM** may use the optimum settings for the MSC SPM packets. The MSC SPM packets all have error detection in the form of a per packet CRC.

This research has shown that the best configuration for this standard is typically when the MSC Datagroup maximum payload size is set between about 500 and 1,500 bytes.

**Streamed data over MSC EPM** utilises a FEC mechanism capable of correcting up to 96 bytes in error in any 2,472 FEC frame.

This research showed that the additional overhead of the RS data was detrimental to the channel performance when compared with the direct equivalent MSC SPM channel, although this was not a hugely significant effect. Typically the best configuration for this standard is when the MSC Datagroup maximum payload size is set between 500 and 1,500 bytes.
Streamed data over MPEG2-TS/MSC Stream Mode utilises a FEC mechanism capable of correcting up to 8 bytes in error, for every 204 bytes sent, in addition to a convolutional interleaver to disperse burst errors across as many RS FEC blocks as possible.

This research showed that the addition of the this interleaver had the affect of substantially increasing the performance of the RS (204,188) FEC mechanism as the best configuration is with a typical maximum PDU payload size close to 10,000 bytes.

Streamed data over MPE-FEC utilises a stronger FEC mechanism than is offered in either MPEG2-TS/MSC Stream Mode or MSC EPM data broadcast standards, allowing up to either 8,192, 16,384, 24,576 or 32,768 bytes to be corrected in any 65,520, 130,560, 195,840 or 261,120 bytes respectively.

This research found the best configuration for MPE-FEC data channels to be where the maximum IP packet payload size to between 500 and 1,500 bytes.

Carousel data over MSC SPM has the option to repeat data in addition to that the mechanisms provided in streamed data over MSC SPM. This is the most powerful reception-assurance mechanism for MSC SPM based channels, and the strength of repetition was mentioned in other research as mentioned in section 2.2.2.

This research has shown the most optimum channel configuration for carousel data sent over MSC SPM standards on real channels to be a maximum MSC Datagroup size of about 1,000 bytes.

Carousel data over MSC EPM has the option to repeat data, in addition to the mechanisms provided by streamed data over MSC EPM based standards.

This research has shown that the strength of error correction given by repetition is second to the inclusion of FEC mechanisms, as repetitions have a worse affect on the channel performance than the overhead required by the FEC mechanism.

This research has additionally shown that real error profiles means that, when MSC EPM is configured with maximum MSC Datagroup payload sizes of between 500 and 1,500 bytes, it will degrade gracefully for MSC SPM only receivers.

Carousel data over MPE-FEC has the option to repeat data, in addition to the mechanisms provided by streamed data over MPE-FEC based standards.
This research has shown mathematically that the stronger FEC mechanism provided in MPE-FEC when compared with MSC EPM, and the reduced level of overhead by using large MPE with proportionally less overhead than MSC SPM packets, mean that the MPE-FEC protocol tends to have a better channel performance than the equivalent MSC EPM standard for extremely large objects. This should be tempered against the relative bandwidths, where it is clear that a DVB-h transmission has more bandwidth available to it than a DAB ensemble.

This research has also shown that when applied to the errors encountered on a real DAB receiver that MPE-FEC normally performs in a similar manner to 24 byte MSC SPM packets over MSC EPM.

5.2 Further Work

The development process for DAB started in 1988. DMB is built on the same channel technology as DAB. The expectation for bandwidth now is far greater than it was in the mid 1980s. In the future, the channels need to be of greater bandwidth, and technologies such as DVB-h are promising such an improvement. There will always be a limitation of physical bandwidth, so finding mechanisms to increase channel performance is important.

The captured data used in this research was captured from a DAB multiplex. The technology needed to capture a DVB-h multiplex did not exist during this research. It would be an interesting continuation of this research to perform the same experiments on the captured data of a DVB-h multiplex, which would give a better like-for-like comparison of the standards, and a clearer understanding of DVB-h tuning.

Linear encapsulation is employed in most contemporary transport mechanisms. It is unsurprising that DAB packet mode data broadcasting mechanisms are all based on linear encapsulation, as they are easy to implement, and crucially have a small impact on the channel delay. We saw in section 2.2.4 that there are alternative encapsulation techniques that may be better for the broadcast digital radio standards, and certainly worth exploring.

LT codes [43] appear to be good for data services; FLUTE which is included in the DVB-h standard can employ similar techniques. When receiving cached data, the receiver is likely to receive a collection of PDUs in error amongst a collection of PDUs received correctly. If these PDUs could be redefined to contain information about what the payload contains, it is conceivable that there would be enough data to retrieve the files more optimally.
If a new non-linear mechanism were to be implemented, all existing receivers would cease to function as there is no trivial way to make such a mechanism backward-compatible. Choosing the most optimum non-linear mechanism may prove financially or politically difficult as some standards are protected by patents, LT codes being an example. Using a protected standard would mean that an additional royalty will have to be paid, and the consumer ultimately would pay the price.

It has been shown that employing a more predictable selection of data units reduces the cost of the mechanism for storage and processing, whilst having a similar performance [29]. This study was on traditional duplex networks. It would be interesting to make a direct comparison of different encapsulation techniques over broadcast digital radio channels.
References


[27] Google. Route of the error profile capture, from Google maps. http://maps.google.co.uk/maps?ie=UTF8&hl=en&ll=52.173511,0.217667&spn=0.270347,0.450439&sz=11&pw=2, may 2008.


Appendix A

Additional Results

A.1 Performance Metric

A.1.1 Performance of a reliable channel, including transmission of overhead

<table>
<thead>
<tr>
<th>$P_t$ (bytes)</th>
<th>$O$ (bytes)</th>
<th>$P_r$ (bytes)</th>
<th>$k$</th>
<th>$E$ (%)</th>
<th>$Q$ (%)</th>
<th>$C$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1000</td>
<td>100</td>
<td>1</td>
<td>9.090909091</td>
<td>100</td>
<td>9.090909091</td>
</tr>
<tr>
<td>200</td>
<td>1000</td>
<td>200</td>
<td>1</td>
<td>16.666666667</td>
<td>100</td>
<td>16.666666667</td>
</tr>
<tr>
<td>300</td>
<td>1000</td>
<td>300</td>
<td>1</td>
<td>23.07692308</td>
<td>100</td>
<td>23.07692308</td>
</tr>
<tr>
<td>400</td>
<td>1000</td>
<td>400</td>
<td>1</td>
<td>28.57142857</td>
<td>100</td>
<td>28.57142857</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
<td>500</td>
<td>1</td>
<td>33.33333333</td>
<td>100</td>
<td>33.33333333</td>
</tr>
<tr>
<td>600</td>
<td>1000</td>
<td>600</td>
<td>1</td>
<td>37.5</td>
<td>100</td>
<td>37.5</td>
</tr>
<tr>
<td>700</td>
<td>1000</td>
<td>700</td>
<td>1</td>
<td>41.17647059</td>
<td>100</td>
<td>41.17647059</td>
</tr>
<tr>
<td>800</td>
<td>1000</td>
<td>800</td>
<td>1</td>
<td>44.444444444</td>
<td>100</td>
<td>44.444444444</td>
</tr>
<tr>
<td>900</td>
<td>1000</td>
<td>900</td>
<td>1</td>
<td>47.36842105</td>
<td>100</td>
<td>47.36842105</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1</td>
<td>50</td>
<td>100</td>
<td>50</td>
</tr>
</tbody>
</table>

Table A.1: Reliable Performance Results. See figure 4.1
A.1.2 Performance of a lossy channel with reliable reception after two transmissions, including transmission of overhead

<table>
<thead>
<tr>
<th>$P_t$</th>
<th>$O$</th>
<th>$P_r$</th>
<th>$k$</th>
<th>$E$</th>
<th>$Q$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bytes)</td>
<td>(bytes)</td>
<td>(bytes)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td>100</td>
<td>2</td>
<td>9.090909091</td>
<td>50</td>
<td>4.545454545</td>
</tr>
<tr>
<td>200</td>
<td>1000</td>
<td>200</td>
<td>2</td>
<td>16.666666667</td>
<td>50</td>
<td>8.333333333</td>
</tr>
<tr>
<td>300</td>
<td>1000</td>
<td>300</td>
<td>2</td>
<td>23.07692308</td>
<td>50</td>
<td>11.53846154</td>
</tr>
<tr>
<td>400</td>
<td>1000</td>
<td>400</td>
<td>2</td>
<td>28.57142857</td>
<td>50</td>
<td>14.28571429</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
<td>500</td>
<td>2</td>
<td>33.33333333</td>
<td>50</td>
<td>16.66666667</td>
</tr>
<tr>
<td>600</td>
<td>1000</td>
<td>600</td>
<td>2</td>
<td>37.5</td>
<td>50</td>
<td>18.75</td>
</tr>
<tr>
<td>700</td>
<td>1000</td>
<td>700</td>
<td>2</td>
<td>41.17647059</td>
<td>50</td>
<td>20.58823529</td>
</tr>
<tr>
<td>800</td>
<td>1000</td>
<td>800</td>
<td>2</td>
<td>44.444444444</td>
<td>50</td>
<td>22.22222222</td>
</tr>
<tr>
<td>900</td>
<td>1000</td>
<td>900</td>
<td>2</td>
<td>47.36842105</td>
<td>50</td>
<td>23.68421053</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>2</td>
<td>50</td>
<td>50</td>
<td>25</td>
</tr>
</tbody>
</table>

Table A.2: Reliable Performance Results after two transmissions. See figure 4.2

A.1.3 Performance of a lossy channel after two transmissions, including transmission of overhead

<table>
<thead>
<tr>
<th>$P_t$</th>
<th>$O$</th>
<th>$P_r$</th>
<th>$k$</th>
<th>$E$</th>
<th>$Q$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bytes)</td>
<td>(bytes)</td>
<td>(bytes)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td>0</td>
<td>2</td>
<td>9.090909091</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>1000</td>
<td>100</td>
<td>2</td>
<td>16.666666667</td>
<td>25</td>
<td>4.166666667</td>
</tr>
<tr>
<td>300</td>
<td>1000</td>
<td>100</td>
<td>2</td>
<td>23.07692308</td>
<td>16.66666667</td>
<td>3.846153846</td>
</tr>
<tr>
<td>400</td>
<td>1000</td>
<td>200</td>
<td>2</td>
<td>28.57142857</td>
<td>25</td>
<td>7.142857143</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
<td>200</td>
<td>2</td>
<td>33.33333333</td>
<td>20</td>
<td>6.666666667</td>
</tr>
<tr>
<td>600</td>
<td>1000</td>
<td>300</td>
<td>2</td>
<td>37.5</td>
<td>25</td>
<td>9.375</td>
</tr>
<tr>
<td>700</td>
<td>1000</td>
<td>300</td>
<td>2</td>
<td>41.17647059</td>
<td>21.42857143</td>
<td>8.823529412</td>
</tr>
<tr>
<td>800</td>
<td>1000</td>
<td>400</td>
<td>2</td>
<td>44.444444444</td>
<td>25</td>
<td>11.11111111</td>
</tr>
<tr>
<td>900</td>
<td>1000</td>
<td>400</td>
<td>2</td>
<td>47.36842105</td>
<td>22.22222222</td>
<td>10.52631579</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>500</td>
<td>2</td>
<td>50</td>
<td>25</td>
<td>12.5</td>
</tr>
</tbody>
</table>

Table A.3: Lossy Performance Results after two transmissions. See figure 4.3
A.1.4 Performance of a lossy channel needing increasing transmissions, including transmission of overhead

<table>
<thead>
<tr>
<th>$P_t$ (bytes)</th>
<th>$O$ (bytes)</th>
<th>$P_r$ (bytes)</th>
<th>$k$</th>
<th>$E$ (%)</th>
<th>$Q$ (%)</th>
<th>$C$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1000</td>
<td>100</td>
<td>2</td>
<td>9.090909091</td>
<td>50</td>
<td>4.545454545</td>
</tr>
<tr>
<td>200</td>
<td>1000</td>
<td>200</td>
<td>3</td>
<td>16.66666667</td>
<td>33.33333333</td>
<td>5.555555556</td>
</tr>
<tr>
<td>300</td>
<td>1000</td>
<td>300</td>
<td>4</td>
<td>23.07692308</td>
<td>25</td>
<td>5.769230769</td>
</tr>
<tr>
<td>400</td>
<td>1000</td>
<td>400</td>
<td>5</td>
<td>28.57142857</td>
<td>20</td>
<td>5.714285714</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
<td>500</td>
<td>6</td>
<td>33.33333333</td>
<td>16.66666667</td>
<td>5.555555556</td>
</tr>
<tr>
<td>600</td>
<td>1000</td>
<td>600</td>
<td>7</td>
<td>37.5</td>
<td>14.28571429</td>
<td>5.357142857</td>
</tr>
<tr>
<td>700</td>
<td>1000</td>
<td>700</td>
<td>8</td>
<td>41.17647059</td>
<td>12.5</td>
<td>5.147058824</td>
</tr>
<tr>
<td>800</td>
<td>1000</td>
<td>800</td>
<td>9</td>
<td>44.44444444</td>
<td>11.11111111</td>
<td>4.938271605</td>
</tr>
<tr>
<td>900</td>
<td>1000</td>
<td>900</td>
<td>10</td>
<td>47.36842105</td>
<td>10</td>
<td>4.736842105</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>11</td>
<td>50</td>
<td>9.090909091</td>
<td>4.545454545</td>
</tr>
</tbody>
</table>

Table A.4: Lossy Performance Results after multiple transmissions. See figure 4.4
A.2 Mathematical Model

A.2.1 DAB using SPM

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>33.33</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>56.25</td>
</tr>
<tr>
<td>46</td>
<td>3</td>
<td>63.89</td>
</tr>
<tr>
<td>65</td>
<td>4</td>
<td>67.71</td>
</tr>
<tr>
<td>84</td>
<td>5</td>
<td>70.00</td>
</tr>
<tr>
<td>103</td>
<td>6</td>
<td>71.53</td>
</tr>
<tr>
<td>122</td>
<td>7</td>
<td>72.62</td>
</tr>
<tr>
<td>141</td>
<td>8</td>
<td>73.44</td>
</tr>
<tr>
<td>160</td>
<td>9</td>
<td>74.07</td>
</tr>
<tr>
<td>179</td>
<td>10</td>
<td>74.58</td>
</tr>
<tr>
<td>8,178</td>
<td>431</td>
<td>79.06</td>
</tr>
</tbody>
</table>

Table A.5: Best-case Overhead Quantity for 24 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1</td>
<td>66.67</td>
</tr>
<tr>
<td>75</td>
<td>2</td>
<td>78.12</td>
</tr>
<tr>
<td>118</td>
<td>3</td>
<td>81.94</td>
</tr>
<tr>
<td>161</td>
<td>4</td>
<td>83.85</td>
</tr>
<tr>
<td>204</td>
<td>5</td>
<td>85.00</td>
</tr>
<tr>
<td>247</td>
<td>6</td>
<td>85.76</td>
</tr>
<tr>
<td>290</td>
<td>7</td>
<td>86.31</td>
</tr>
<tr>
<td>333</td>
<td>8</td>
<td>86.72</td>
</tr>
<tr>
<td>376</td>
<td>9</td>
<td>87.04</td>
</tr>
<tr>
<td>419</td>
<td>10</td>
<td>87.29</td>
</tr>
<tr>
<td>8,159</td>
<td>190</td>
<td>89.46</td>
</tr>
</tbody>
</table>

Table A.6: Best-case Overhead Quantity for 48 byte SPM packets
### Table A.7: Best-case Overhead Quantity for 72 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>1</td>
<td>77.78</td>
</tr>
<tr>
<td>123</td>
<td>2</td>
<td>85.42</td>
</tr>
<tr>
<td>190</td>
<td>3</td>
<td>87.96</td>
</tr>
<tr>
<td>257</td>
<td>4</td>
<td>89.24</td>
</tr>
<tr>
<td>324</td>
<td>5</td>
<td>90.00</td>
</tr>
<tr>
<td>391</td>
<td>6</td>
<td>90.51</td>
</tr>
<tr>
<td>458</td>
<td>7</td>
<td>90.87</td>
</tr>
<tr>
<td>525</td>
<td>8</td>
<td>91.15</td>
</tr>
<tr>
<td>592</td>
<td>9</td>
<td>91.36</td>
</tr>
<tr>
<td>659</td>
<td>10</td>
<td>91.53</td>
</tr>
<tr>
<td>8,163</td>
<td>122</td>
<td>92.93</td>
</tr>
</tbody>
</table>

### Table A.8: Best-case Overhead Quantity for 96 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>1</td>
<td>83.33</td>
</tr>
<tr>
<td>171</td>
<td>2</td>
<td>89.06</td>
</tr>
<tr>
<td>262</td>
<td>3</td>
<td>90.97</td>
</tr>
<tr>
<td>353</td>
<td>4</td>
<td>91.93</td>
</tr>
<tr>
<td>444</td>
<td>5</td>
<td>92.50</td>
</tr>
<tr>
<td>535</td>
<td>6</td>
<td>92.88</td>
</tr>
<tr>
<td>626</td>
<td>7</td>
<td>93.15</td>
</tr>
<tr>
<td>717</td>
<td>8</td>
<td>93.36</td>
</tr>
<tr>
<td>808</td>
<td>9</td>
<td>93.52</td>
</tr>
<tr>
<td>899</td>
<td>10</td>
<td>93.65</td>
</tr>
<tr>
<td>8,088</td>
<td>89</td>
<td>94.66</td>
</tr>
</tbody>
</table>
A.2.2 DAB using EPM

Table A.9: Best-case Overhead Quantity for 24 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>0.32</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>1.08</td>
</tr>
<tr>
<td>46</td>
<td>3</td>
<td>1.83</td>
</tr>
<tr>
<td>65</td>
<td>4</td>
<td>2.56</td>
</tr>
<tr>
<td>84</td>
<td>5</td>
<td>3.29</td>
</tr>
<tr>
<td>103</td>
<td>6</td>
<td>4.00</td>
</tr>
<tr>
<td>122</td>
<td>7</td>
<td>4.70</td>
</tr>
<tr>
<td>141</td>
<td>8</td>
<td>5.40</td>
</tr>
<tr>
<td>160</td>
<td>9</td>
<td>6.08</td>
</tr>
<tr>
<td>179</td>
<td>10</td>
<td>6.75</td>
</tr>
<tr>
<td>8,178</td>
<td>431</td>
<td>39.82</td>
</tr>
</tbody>
</table>

Table A.10: Best-case Overhead Quantity for 48 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1</td>
<td>1.28</td>
</tr>
<tr>
<td>75</td>
<td>2</td>
<td>2.94</td>
</tr>
<tr>
<td>118</td>
<td>3</td>
<td>4.56</td>
</tr>
<tr>
<td>161</td>
<td>4</td>
<td>6.11</td>
</tr>
<tr>
<td>204</td>
<td>5</td>
<td>7.62</td>
</tr>
<tr>
<td>247</td>
<td>6</td>
<td>9.08</td>
</tr>
<tr>
<td>290</td>
<td>7</td>
<td>10.50</td>
</tr>
<tr>
<td>333</td>
<td>8</td>
<td>11.87</td>
</tr>
<tr>
<td>376</td>
<td>9</td>
<td>13.20</td>
</tr>
<tr>
<td>419</td>
<td>10</td>
<td>14.49</td>
</tr>
<tr>
<td>8,159</td>
<td>190</td>
<td>39.76</td>
</tr>
</tbody>
</table>

A.2.3 IP DAB using SPM

A.2.4 IP DAB using EPM

A.3 Simulation Results
Table A.11: Best-case Overhead Quantity for 72 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>1</td>
<td>2.22</td>
</tr>
<tr>
<td>123</td>
<td>2</td>
<td>4.74</td>
</tr>
<tr>
<td>190</td>
<td>3</td>
<td>7.14</td>
</tr>
<tr>
<td>257</td>
<td>4</td>
<td>9.42</td>
</tr>
<tr>
<td>324</td>
<td>5</td>
<td>11.59</td>
</tr>
<tr>
<td>391</td>
<td>6</td>
<td>13.66</td>
</tr>
<tr>
<td>458</td>
<td>7</td>
<td>15.63</td>
</tr>
<tr>
<td>525</td>
<td>8</td>
<td>17.52</td>
</tr>
<tr>
<td>592</td>
<td>9</td>
<td>19.32</td>
</tr>
<tr>
<td>659</td>
<td>10</td>
<td>21.05</td>
</tr>
<tr>
<td>8,163</td>
<td>122</td>
<td>45.22</td>
</tr>
</tbody>
</table>

Table A.12: Best-case Overhead Quantity for 96 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>1</td>
<td>3.13</td>
</tr>
<tr>
<td>171</td>
<td>2</td>
<td>6.47</td>
</tr>
<tr>
<td>262</td>
<td>3</td>
<td>9.58</td>
</tr>
<tr>
<td>353</td>
<td>4</td>
<td>12.50</td>
</tr>
<tr>
<td>444</td>
<td>5</td>
<td>15.23</td>
</tr>
<tr>
<td>535</td>
<td>6</td>
<td>17.79</td>
</tr>
<tr>
<td>626</td>
<td>7</td>
<td>20.21</td>
</tr>
<tr>
<td>717</td>
<td>8</td>
<td>22.48</td>
</tr>
<tr>
<td>808</td>
<td>9</td>
<td>24.63</td>
</tr>
<tr>
<td>899</td>
<td>10</td>
<td>26.67</td>
</tr>
<tr>
<td>8,088</td>
<td>89</td>
<td>44.99</td>
</tr>
</tbody>
</table>

Table A.13: Best-case Overhead Quantity for 24 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>14.58</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>36.11</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>46.88</td>
</tr>
<tr>
<td>64</td>
<td>4</td>
<td>53.33</td>
</tr>
<tr>
<td>83</td>
<td>5</td>
<td>57.64</td>
</tr>
<tr>
<td>102</td>
<td>6</td>
<td>60.71</td>
</tr>
<tr>
<td>121</td>
<td>7</td>
<td>63.02</td>
</tr>
<tr>
<td>140</td>
<td>8</td>
<td>64.81</td>
</tr>
<tr>
<td>159</td>
<td>9</td>
<td>66.25</td>
</tr>
<tr>
<td>178</td>
<td>10</td>
<td>67.42</td>
</tr>
<tr>
<td>8,177</td>
<td>432</td>
<td>78.87</td>
</tr>
<tr>
<td>Datagroup payload</td>
<td>Number of SPM packets needed</td>
<td>Performance (%)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>57.29</td>
</tr>
<tr>
<td>98</td>
<td>3</td>
<td>68.06</td>
</tr>
<tr>
<td>141</td>
<td>4</td>
<td>73.44</td>
</tr>
<tr>
<td>184</td>
<td>5</td>
<td>76.67</td>
</tr>
<tr>
<td>227</td>
<td>6</td>
<td>78.82</td>
</tr>
<tr>
<td>270</td>
<td>7</td>
<td>80.36</td>
</tr>
<tr>
<td>313</td>
<td>8</td>
<td>81.51</td>
</tr>
<tr>
<td>356</td>
<td>9</td>
<td>82.41</td>
</tr>
<tr>
<td>399</td>
<td>10</td>
<td>83.13</td>
</tr>
<tr>
<td>8,182</td>
<td>191</td>
<td>89.25</td>
</tr>
</tbody>
</table>

Table A.14: Best-case Overhead Quantity for 48 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>103</td>
<td>2</td>
<td>71.53</td>
</tr>
<tr>
<td>170</td>
<td>3</td>
<td>78.70</td>
</tr>
<tr>
<td>237</td>
<td>4</td>
<td>82.29</td>
</tr>
<tr>
<td>304</td>
<td>5</td>
<td>84.44</td>
</tr>
<tr>
<td>371</td>
<td>6</td>
<td>85.88</td>
</tr>
<tr>
<td>438</td>
<td>7</td>
<td>86.90</td>
</tr>
<tr>
<td>505</td>
<td>8</td>
<td>87.67</td>
</tr>
<tr>
<td>572</td>
<td>9</td>
<td>88.27</td>
</tr>
<tr>
<td>639</td>
<td>10</td>
<td>88.75</td>
</tr>
<tr>
<td>8,143</td>
<td>122</td>
<td>92.70</td>
</tr>
</tbody>
</table>

Table A.15: Best-case Overhead Quantity for 72 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>1</td>
<td>62.5</td>
</tr>
<tr>
<td>151</td>
<td>2</td>
<td>78.65</td>
</tr>
<tr>
<td>242</td>
<td>3</td>
<td>84.03</td>
</tr>
<tr>
<td>333</td>
<td>4</td>
<td>86.72</td>
</tr>
<tr>
<td>424</td>
<td>5</td>
<td>88.33</td>
</tr>
<tr>
<td>515</td>
<td>6</td>
<td>89.41</td>
</tr>
<tr>
<td>606</td>
<td>7</td>
<td>90.18</td>
</tr>
<tr>
<td>697</td>
<td>8</td>
<td>90.76</td>
</tr>
<tr>
<td>788</td>
<td>9</td>
<td>91.20</td>
</tr>
<tr>
<td>879</td>
<td>10</td>
<td>91.56</td>
</tr>
<tr>
<td>8,159</td>
<td>90</td>
<td>94.43</td>
</tr>
</tbody>
</table>

Table A.16: Best-case Overhead Quantity for 96 byte SPM packets
<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>0.28</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>1.04</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>1.79</td>
</tr>
<tr>
<td>64</td>
<td>4</td>
<td>2.52</td>
</tr>
<tr>
<td>83</td>
<td>5</td>
<td>3.25</td>
</tr>
<tr>
<td>102</td>
<td>6</td>
<td>3.96</td>
</tr>
<tr>
<td>121</td>
<td>7</td>
<td>4.67</td>
</tr>
<tr>
<td>140</td>
<td>8</td>
<td>5.36</td>
</tr>
<tr>
<td>159</td>
<td>9</td>
<td>6.04</td>
</tr>
<tr>
<td>178</td>
<td>10</td>
<td>6.72</td>
</tr>
<tr>
<td>8,177</td>
<td>432</td>
<td>39.82</td>
</tr>
</tbody>
</table>

Table A.17: Best-case Overhead Quantity for 24 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>0.48</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>2.18</td>
</tr>
<tr>
<td>98</td>
<td>3</td>
<td>3.81</td>
</tr>
<tr>
<td>141</td>
<td>4</td>
<td>5.4</td>
</tr>
<tr>
<td>184</td>
<td>5</td>
<td>6.93</td>
</tr>
<tr>
<td>227</td>
<td>6</td>
<td>8.41</td>
</tr>
<tr>
<td>270</td>
<td>7</td>
<td>9.85</td>
</tr>
<tr>
<td>313</td>
<td>8</td>
<td>11.24</td>
</tr>
<tr>
<td>356</td>
<td>9</td>
<td>12.59</td>
</tr>
<tr>
<td>399</td>
<td>10</td>
<td>13.9</td>
</tr>
<tr>
<td>8,182</td>
<td>191</td>
<td>39.83</td>
</tr>
</tbody>
</table>

Table A.18: Best-case Overhead Quantity for 48 byte SPM packets

<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>1</td>
<td>1.44</td>
</tr>
<tr>
<td>103</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>170</td>
<td>3</td>
<td>6.43</td>
</tr>
<tr>
<td>237</td>
<td>4</td>
<td>8.75</td>
</tr>
<tr>
<td>304</td>
<td>5</td>
<td>10.95</td>
</tr>
<tr>
<td>371</td>
<td>6</td>
<td>13.05</td>
</tr>
<tr>
<td>438</td>
<td>7</td>
<td>15.05</td>
</tr>
<tr>
<td>505</td>
<td>8</td>
<td>16.96</td>
</tr>
<tr>
<td>572</td>
<td>9</td>
<td>18.79</td>
</tr>
<tr>
<td>639</td>
<td>10</td>
<td>20.54</td>
</tr>
<tr>
<td>8,143</td>
<td>122</td>
<td>45.16</td>
</tr>
</tbody>
</table>

Table A.19: Best-case Overhead Quantity for 72 byte SPM packets
<table>
<thead>
<tr>
<th>Datagroup payload</th>
<th>Number of SPM packets needed</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>1</td>
<td>2.37</td>
</tr>
<tr>
<td>151</td>
<td>2</td>
<td>5.76</td>
</tr>
<tr>
<td>242</td>
<td>3</td>
<td>8.92</td>
</tr>
<tr>
<td>333</td>
<td>4</td>
<td>11.87</td>
</tr>
<tr>
<td>424</td>
<td>5</td>
<td>14.64</td>
</tr>
<tr>
<td>515</td>
<td>6</td>
<td>17.24</td>
</tr>
<tr>
<td>606</td>
<td>7</td>
<td>19.69</td>
</tr>
<tr>
<td>697</td>
<td>8</td>
<td>21.99</td>
</tr>
<tr>
<td>788</td>
<td>9</td>
<td>24.17</td>
</tr>
<tr>
<td>879</td>
<td>10</td>
<td>26.23</td>
</tr>
<tr>
<td>8,159</td>
<td>90</td>
<td>45.21</td>
</tr>
</tbody>
</table>

Table A.20: Best-case Overhead Quantity for 96 byte SPM packets

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>62.18</td>
</tr>
<tr>
<td>1256</td>
<td>62.02</td>
</tr>
<tr>
<td>1330</td>
<td><strong>62.54</strong></td>
</tr>
<tr>
<td>1446</td>
<td>62.24</td>
</tr>
<tr>
<td>1500</td>
<td>62.04</td>
</tr>
<tr>
<td>5015</td>
<td>38.75</td>
</tr>
</tbody>
</table>

Table A.21: Select results for streamed object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 1. See figure 4.73a

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td></td>
</tr>
<tr>
<td>670</td>
<td>59.30 67.89 73.25 69.12</td>
</tr>
<tr>
<td>1000</td>
<td>61.05 70.54 69.30 71.64</td>
</tr>
<tr>
<td>1256</td>
<td>61.79 71.48 70.45 73.00</td>
</tr>
<tr>
<td>1446</td>
<td>62.15 71.88 70.01 73.54</td>
</tr>
<tr>
<td>1500</td>
<td>62.14 72.09 69.92 73.69</td>
</tr>
<tr>
<td>1677</td>
<td>62.44 72.36 69.87 74.10</td>
</tr>
<tr>
<td>3192</td>
<td>63.11 67.08 71.40 75.11</td>
</tr>
<tr>
<td>5005</td>
<td>62.73 66.94 71.24 75.33</td>
</tr>
<tr>
<td>5015</td>
<td>62.73 66.94 71.24 75.33</td>
</tr>
</tbody>
</table>

Table A.22: Select results for streamed object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 2 part 1. See figure 4.73b
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>456</td>
<td>60.66</td>
</tr>
<tr>
<td>737</td>
<td>59.24</td>
</tr>
<tr>
<td>1000</td>
<td>60.10</td>
</tr>
<tr>
<td>1256</td>
<td>60.45</td>
</tr>
<tr>
<td>1446</td>
<td>60.35</td>
</tr>
<tr>
<td>1500</td>
<td>60.54</td>
</tr>
<tr>
<td>1677</td>
<td>60.49</td>
</tr>
<tr>
<td>1729</td>
<td>60.47</td>
</tr>
<tr>
<td>5015</td>
<td>54.23</td>
</tr>
</tbody>
</table>

Table A.23: Select results for streamed object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 2 part 2. See figure 4.73c

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>402</td>
<td>60.35</td>
</tr>
<tr>
<td>418</td>
<td><strong>60.80</strong></td>
</tr>
<tr>
<td>731</td>
<td>54.99</td>
</tr>
<tr>
<td>1000</td>
<td>55.97</td>
</tr>
<tr>
<td>1256</td>
<td>56.37</td>
</tr>
<tr>
<td>1446</td>
<td>56.41</td>
</tr>
<tr>
<td>1500</td>
<td>56.47</td>
</tr>
<tr>
<td>2093</td>
<td>56.42</td>
</tr>
<tr>
<td>5015</td>
<td>54.49</td>
</tr>
</tbody>
</table>

Table A.24: Select results for streamed object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 3 part 1. See figure 4.73d
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>301</td>
<td>47.39</td>
</tr>
<tr>
<td>364</td>
<td>48.64</td>
</tr>
<tr>
<td>402</td>
<td>49.11</td>
</tr>
<tr>
<td>418</td>
<td>49.24</td>
</tr>
<tr>
<td>1000</td>
<td>45.70</td>
</tr>
<tr>
<td>1256</td>
<td>44.85</td>
</tr>
<tr>
<td>1446</td>
<td>44.39</td>
</tr>
<tr>
<td>1500</td>
<td>44.16</td>
</tr>
<tr>
<td>5015</td>
<td>36.01</td>
</tr>
</tbody>
</table>

Table A.25: Select results for streamed object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 3 part 2. See figure 4.73e
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>48</td>
</tr>
<tr>
<td>1000</td>
<td>64.75</td>
</tr>
<tr>
<td>1256</td>
<td>65.47</td>
</tr>
<tr>
<td>1368</td>
<td>65.71</td>
</tr>
<tr>
<td>1446</td>
<td>65.01</td>
</tr>
<tr>
<td>1500</td>
<td>65.08</td>
</tr>
<tr>
<td>1541</td>
<td>65.14</td>
</tr>
<tr>
<td>1729</td>
<td>65.33</td>
</tr>
<tr>
<td>2107</td>
<td>65.46</td>
</tr>
<tr>
<td>5015</td>
<td>64.57</td>
</tr>
</tbody>
</table>

Table A.26: Select results for streamed object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 4. See figure 4.73f

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>54.46</td>
</tr>
<tr>
<td>1256</td>
<td>55.48</td>
</tr>
<tr>
<td>1446</td>
<td>55.84</td>
</tr>
<tr>
<td>1500</td>
<td>55.87</td>
</tr>
<tr>
<td>3487</td>
<td>57.22</td>
</tr>
<tr>
<td>5015</td>
<td>56.31</td>
</tr>
</tbody>
</table>

Table A.27: Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 1. See figure 4.76a
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>603</td>
<td>54.34</td>
</tr>
<tr>
<td>1000</td>
<td>57.42</td>
</tr>
<tr>
<td>1256</td>
<td>58.28</td>
</tr>
<tr>
<td>1419</td>
<td>58.86</td>
</tr>
<tr>
<td>1446</td>
<td>58.86</td>
</tr>
<tr>
<td>1500</td>
<td>59.06</td>
</tr>
<tr>
<td>3525</td>
<td>60.97</td>
</tr>
<tr>
<td>5015</td>
<td>54.61</td>
</tr>
<tr>
<td>6370</td>
<td>54.66</td>
</tr>
</tbody>
</table>

Table A.28: Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 2 part 1. See figure 4.76b

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>946</td>
<td>56.77</td>
</tr>
<tr>
<td>1000</td>
<td>56.94</td>
</tr>
<tr>
<td>1092</td>
<td>57.24</td>
</tr>
<tr>
<td>1256</td>
<td>57.75</td>
</tr>
<tr>
<td>1446</td>
<td>58.20</td>
</tr>
<tr>
<td>1500</td>
<td>58.44</td>
</tr>
<tr>
<td>2747</td>
<td>59.49</td>
</tr>
<tr>
<td>3525</td>
<td>60.17</td>
</tr>
<tr>
<td>5015</td>
<td>59.21</td>
</tr>
</tbody>
</table>

Table A.29: Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 2 part 2. See figure 4.76c
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>56.96</td>
</tr>
<tr>
<td>1256</td>
<td>57.89</td>
</tr>
<tr>
<td>1446</td>
<td>58.39</td>
</tr>
<tr>
<td>1500</td>
<td>58.62</td>
</tr>
<tr>
<td>2093</td>
<td>59.35</td>
</tr>
<tr>
<td>4154</td>
<td>60.29</td>
</tr>
<tr>
<td>5015</td>
<td>60.34</td>
</tr>
<tr>
<td>5117</td>
<td>60.23</td>
</tr>
<tr>
<td>5311</td>
<td><strong>60.87</strong></td>
</tr>
</tbody>
</table>

Table A.30: Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 3 part 1. See figure 4.76d
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>51.17</td>
</tr>
<tr>
<td>1075</td>
<td>51.44</td>
</tr>
<tr>
<td>1256</td>
<td>51.77</td>
</tr>
<tr>
<td>1446</td>
<td>52.09</td>
</tr>
<tr>
<td>1500</td>
<td>52.20</td>
</tr>
<tr>
<td>1739</td>
<td>53.99</td>
</tr>
<tr>
<td>2093</td>
<td>52.40</td>
</tr>
<tr>
<td>4154</td>
<td>51.22</td>
</tr>
<tr>
<td>5015</td>
<td>50.82</td>
</tr>
</tbody>
</table>

Table A.31: Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 3 part 2. See figure 4.76e

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>61.62</td>
</tr>
<tr>
<td>1256</td>
<td>61.87</td>
</tr>
<tr>
<td>1446</td>
<td>62.48</td>
</tr>
<tr>
<td>1500</td>
<td>62.69</td>
</tr>
<tr>
<td>3283</td>
<td>63.13</td>
</tr>
<tr>
<td>5015</td>
<td>63.39</td>
</tr>
<tr>
<td>6370</td>
<td>63.43</td>
</tr>
<tr>
<td>6708</td>
<td>63.58</td>
</tr>
<tr>
<td>7097</td>
<td><strong>63.73</strong></td>
</tr>
</tbody>
</table>

Table A.32: Select results for steam object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 4. See figure 4.76f
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Size Carried By MPEG2-TS/MSC Stream Mode</td>
</tr>
<tr>
<td>1000</td>
<td>23.05</td>
</tr>
<tr>
<td>1256</td>
<td>23.11</td>
</tr>
<tr>
<td>1446</td>
<td>23.17</td>
</tr>
<tr>
<td>1500</td>
<td>23.20</td>
</tr>
<tr>
<td>5015</td>
<td>23.38</td>
</tr>
<tr>
<td>8832</td>
<td>23.41</td>
</tr>
</tbody>
</table>

Table A.33: Select results for 1000000 Object over MPEG2-TS/MSC Stream Mode/Cambridge All Zeros 1. See figure 4.79a

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Size Carried By MPEG2-TS/MSC Stream Mode</td>
</tr>
<tr>
<td>1000</td>
<td>36.02</td>
</tr>
<tr>
<td>1256</td>
<td>36.13</td>
</tr>
<tr>
<td>1446</td>
<td>36.21</td>
</tr>
<tr>
<td>1500</td>
<td>36.26</td>
</tr>
<tr>
<td>5015</td>
<td>36.54</td>
</tr>
<tr>
<td>9936</td>
<td>36.60</td>
</tr>
</tbody>
</table>

Table A.34: Select results for 1000000 Object over MPEG2-TS/MSC Stream Mode/Cambridge All Zeros 2 part 1. See figure 4.79b

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Size Carried By MPEG2-TS/MSC Stream Mode</td>
</tr>
<tr>
<td>184</td>
<td>36.73</td>
</tr>
<tr>
<td>1000</td>
<td>35.39</td>
</tr>
<tr>
<td>1256</td>
<td>35.49</td>
</tr>
<tr>
<td>1446</td>
<td>35.57</td>
</tr>
<tr>
<td>1500</td>
<td>35.62</td>
</tr>
<tr>
<td>5015</td>
<td>35.89</td>
</tr>
</tbody>
</table>

Table A.35: Select results for 1000000 Object over MPEG2-TS/MSC Stream Mode/Cambridge All Zeros 2 part 2. See figure 4.79c
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Size Carried By MPEG2-TS/MSC Stream Mode</td>
</tr>
<tr>
<td>1000</td>
<td>35.56</td>
</tr>
<tr>
<td>1256</td>
<td>35.67</td>
</tr>
<tr>
<td>1446</td>
<td>35.74</td>
</tr>
<tr>
<td>1500</td>
<td>35.80</td>
</tr>
<tr>
<td>5015</td>
<td>36.07</td>
</tr>
<tr>
<td>9568</td>
<td>36.13</td>
</tr>
</tbody>
</table>

Table A.36: Select results for 1000000 Object over MPEG2-TS/MSC Stream Mode/Cambridge All Zeros 3 part 1. See figure 4.79d

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Size Carried By MPEG2-TS/MSC Stream Mode</td>
</tr>
<tr>
<td>1000</td>
<td>31.17</td>
</tr>
<tr>
<td>1256</td>
<td>31.26</td>
</tr>
<tr>
<td>1446</td>
<td>31.33</td>
</tr>
<tr>
<td>1500</td>
<td>31.38</td>
</tr>
<tr>
<td>5015</td>
<td>31.63</td>
</tr>
<tr>
<td>8648</td>
<td>31.68</td>
</tr>
</tbody>
</table>

Table A.37: Select results for 1000000 Object over MPEG2-TS/MSC Stream Mode/Cambridge All Zeros 3 part 2. See figure 4.79e
### Table A.38: Select results for 1000000 Object over MPEG2-TS/MSC Stream Mode/Cambridge All Zeros 4. See figure 4.79f

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Size Carried By MPEG2-TS/MSC Stream Mode</td>
</tr>
<tr>
<td>1000</td>
<td>39.65</td>
</tr>
<tr>
<td>1256</td>
<td>39.76</td>
</tr>
<tr>
<td>1446</td>
<td>39.85</td>
</tr>
<tr>
<td>1472</td>
<td>39.91</td>
</tr>
<tr>
<td>1500</td>
<td>39.91</td>
</tr>
<tr>
<td>5015</td>
<td>39.47</td>
</tr>
</tbody>
</table>

### Table A.39: Select results for streamed object over IP/MSC-FEC/Cambridge All Zeros 1. See figure 4.81a

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td>1000</td>
<td>60.40</td>
</tr>
<tr>
<td>1256</td>
<td>60.54</td>
</tr>
<tr>
<td>1446</td>
<td>65.27</td>
</tr>
<tr>
<td>1500</td>
<td>65.32</td>
</tr>
<tr>
<td>4000</td>
<td>65.70</td>
</tr>
<tr>
<td>5015</td>
<td>65.70</td>
</tr>
</tbody>
</table>

### Table A.40: Select results for streamed object over IP/MSC-FEC/Cambridge All Zeros 2 part 1. See figure 4.81b

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td>1000</td>
<td>62.64</td>
</tr>
<tr>
<td>1200</td>
<td>62.65</td>
</tr>
<tr>
<td>1256</td>
<td>62.65</td>
</tr>
<tr>
<td>1446</td>
<td>61.86</td>
</tr>
<tr>
<td>1500</td>
<td>61.87</td>
</tr>
<tr>
<td>3900</td>
<td>61.99</td>
</tr>
<tr>
<td>4000</td>
<td>61.99</td>
</tr>
<tr>
<td>5015</td>
<td>61.99</td>
</tr>
<tr>
<td>Payload (bytes)</td>
<td>Performance of MPE-FEC row size (%)</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>256</td>
<td>56.71</td>
</tr>
<tr>
<td>512</td>
<td>62.88</td>
</tr>
<tr>
<td>768</td>
<td>62.95</td>
</tr>
<tr>
<td>1000</td>
<td>62.94</td>
</tr>
<tr>
<td>1256</td>
<td>69.13</td>
</tr>
<tr>
<td>1446</td>
<td>69.16</td>
</tr>
<tr>
<td>1500</td>
<td>69.16</td>
</tr>
<tr>
<td>1000</td>
<td>62.82</td>
</tr>
<tr>
<td>1200</td>
<td>62.85</td>
</tr>
<tr>
<td>1256</td>
<td>62.85</td>
</tr>
<tr>
<td>1446</td>
<td>58.91</td>
</tr>
<tr>
<td>1500</td>
<td>58.91</td>
</tr>
<tr>
<td>3500</td>
<td>58.88</td>
</tr>
<tr>
<td>3900</td>
<td>58.88</td>
</tr>
<tr>
<td>4000</td>
<td>58.88</td>
</tr>
<tr>
<td>5015</td>
<td>58.88</td>
</tr>
</tbody>
</table>

Table A.41: Select results for streamed object over IP/MSC-FEC/Cambridge All Zeros 2 part 2. See figure 4.81c.

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>62.82</td>
<td>62.94</td>
<td>53.16</td>
<td>53.52</td>
</tr>
<tr>
<td>1200</td>
<td>62.85</td>
<td>62.96</td>
<td>53.17</td>
<td>53.53</td>
</tr>
<tr>
<td>1256</td>
<td>62.85</td>
<td>62.96</td>
<td>53.17</td>
<td>53.53</td>
</tr>
<tr>
<td>1446</td>
<td>58.91</td>
<td>59.04</td>
<td>53.52</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>58.91</td>
<td>59.03</td>
<td>53.53</td>
<td></td>
</tr>
<tr>
<td>3500</td>
<td>58.88</td>
<td>59.06</td>
<td>53.59</td>
<td></td>
</tr>
<tr>
<td>3900</td>
<td>58.88</td>
<td>63.05</td>
<td>59.06</td>
<td>53.60</td>
</tr>
<tr>
<td>4000</td>
<td>58.88</td>
<td>63.05</td>
<td>59.06</td>
<td>53.60</td>
</tr>
<tr>
<td>5015</td>
<td>58.88</td>
<td>63.05</td>
<td>59.06</td>
<td>53.60</td>
</tr>
</tbody>
</table>

Table A.42: Select results for streamed object over IP/MSC-FEC/Cambridge All Zeros 3 part 1. See figure 4.81d.
| Payload (bytes) | Performance of MPE-FEC row size (%) |  |
|----------------|-----------------------------------|--|---|
|                | 256 | 512 | 768 | 1024 |
| 500            | 61.44 | **61.41** | 56.56 | 56.67 |
| 1000           | 61.37 | 61.34 | 56.62 | 56.77 |
| 1100           | 61.37 | 61.34 | 56.63 | **56.77** |
| 1256           | 61.36 | 61.33 | 56.63 | 56.77 |
| 1300           | **62.24** | 61.32 | **62.31** | 56.77 |
| 1446           | 62.23 | 61.32 | 62.30 | 56.77 |
| 1500           | 62.22 | 61.32 | 62.30 | 56.77 |
| 5015           | 62.16 | 61.29 | 62.25 | 56.76 |

Table A.43: Select results for streamed object over IP/MSC-FEC/Cambridge All Zeros 3 part 2. See figure 4.81e

| Payload (bytes) | Performance of MPE-FEC row size (%) |  |
|----------------|-----------------------------------|--|---|
|                | 256 | 512 | 768 | 1024 |
| 1000           | 65.10 | 65.02 | 59.73 | 59.78 |
| 1256           | 65.10 | 65.02 | 59.74 | 59.78 |
| 1300           | **68.14** | 65.02 | 68.08 | 59.78 |
| 1446           | 68.14 | 65.03 | 68.08 | 59.78 |
| 1500           | 68.14 | 65.03 | 68.08 | 59.78 |
| 3900           | 68.12 | **65.04** | **68.11** | 59.80 |
| 4000           | 68.12 | 65.04 | 68.11 | **59.80** |
| 5015           | 68.12 | 65.04 | 68.11 | **59.80** |

Table A.44: Select results for streamed object over IP/MSC-FEC/Cambridge All Zeros 4. See figure 4.81f
## Payload (bytes) vs Performance of Maximum MSC SPM Packet size (%)

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>3.67</td>
</tr>
<tr>
<td>1256</td>
<td>3.70</td>
</tr>
<tr>
<td>1446</td>
<td>3.70</td>
</tr>
<tr>
<td>1500</td>
<td>3.72</td>
</tr>
<tr>
<td>1584</td>
<td><strong>3.74</strong></td>
</tr>
<tr>
<td>5015</td>
<td>3.61</td>
</tr>
</tbody>
</table>

Table A.45: Select results for carousel object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 1. See figure 4.91a
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>46.03</td>
</tr>
<tr>
<td>1256</td>
<td>46.64</td>
</tr>
<tr>
<td>1446</td>
<td>46.92</td>
</tr>
<tr>
<td>1500</td>
<td>47.26</td>
</tr>
<tr>
<td>4214</td>
<td>34.08</td>
</tr>
<tr>
<td>5015</td>
<td>47.87</td>
</tr>
<tr>
<td>7056</td>
<td>48.07</td>
</tr>
<tr>
<td>7826</td>
<td>46.02</td>
</tr>
<tr>
<td>8107</td>
<td>45.96</td>
</tr>
</tbody>
</table>

Table A.46: Select results for carousel object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 2 part 1. See figure 4.91b

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>182</td>
<td>17.17</td>
</tr>
<tr>
<td>201</td>
<td>21.39</td>
</tr>
<tr>
<td>1000</td>
<td>25.99</td>
</tr>
<tr>
<td>1256</td>
<td>27.26</td>
</tr>
<tr>
<td>1446</td>
<td>26.72</td>
</tr>
<tr>
<td>1500</td>
<td>32.47</td>
</tr>
<tr>
<td>3182</td>
<td>21.16</td>
</tr>
<tr>
<td>3696</td>
<td>34.91</td>
</tr>
<tr>
<td>5015</td>
<td>32.44</td>
</tr>
</tbody>
</table>

Table A.47: Select results for carousel object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 2 part 2. See figure 4.91c
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>288</td>
<td>24.67</td>
</tr>
<tr>
<td>1000</td>
<td>14.45</td>
</tr>
<tr>
<td>1256</td>
<td>14.64</td>
</tr>
<tr>
<td>1274</td>
<td>14.64</td>
</tr>
<tr>
<td>1446</td>
<td>14.73</td>
</tr>
<tr>
<td>1500</td>
<td>14.85</td>
</tr>
<tr>
<td>2747</td>
<td>14.95</td>
</tr>
<tr>
<td>3784</td>
<td>15.02</td>
</tr>
<tr>
<td>5015</td>
<td>15.02</td>
</tr>
</tbody>
</table>

Table A.48: Select results for carousel object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 3 part 1. See figure 4.91d

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>11.24</td>
</tr>
<tr>
<td>1256</td>
<td>9.17</td>
</tr>
<tr>
<td>1446</td>
<td>12.50</td>
</tr>
<tr>
<td>1500</td>
<td>9.68</td>
</tr>
<tr>
<td>1675</td>
<td>11.96</td>
</tr>
<tr>
<td>1728</td>
<td>12.58</td>
</tr>
<tr>
<td>3731</td>
<td>8.45</td>
</tr>
<tr>
<td>5015</td>
<td>7.49</td>
</tr>
<tr>
<td>6106</td>
<td>7.50</td>
</tr>
</tbody>
</table>

Table A.49: Select results for carousel object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 3 part 2. See figure 4.91e
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>52.92</td>
</tr>
<tr>
<td>1256</td>
<td>54.86</td>
</tr>
<tr>
<td>1446</td>
<td>54.75</td>
</tr>
<tr>
<td>1500</td>
<td>55.29</td>
</tr>
<tr>
<td>2448</td>
<td>56.87</td>
</tr>
<tr>
<td>5015</td>
<td>55.97</td>
</tr>
<tr>
<td>5633</td>
<td>55.98</td>
</tr>
<tr>
<td>8099</td>
<td>55.41</td>
</tr>
<tr>
<td>8107</td>
<td>55.41</td>
</tr>
</tbody>
</table>

Table A.50: Select results for carousel object over IP/MSC Datagroups/MSC SPM/Cambridge All Zeros 4. See figure 4.91f
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>3.33</td>
</tr>
<tr>
<td>1256</td>
<td>3.40</td>
</tr>
<tr>
<td>1446</td>
<td>3.41</td>
</tr>
<tr>
<td>1500</td>
<td>3.44</td>
</tr>
<tr>
<td>5015</td>
<td>3.49</td>
</tr>
<tr>
<td>5568</td>
<td>3.55</td>
</tr>
</tbody>
</table>

Table A.51: Select results for carousel object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 1. See figure 4.93a

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>59.46</td>
</tr>
<tr>
<td>1256</td>
<td>61.25</td>
</tr>
<tr>
<td>1446</td>
<td>61.58</td>
</tr>
<tr>
<td>1500</td>
<td>62.11</td>
</tr>
<tr>
<td>5015</td>
<td>62.88</td>
</tr>
<tr>
<td>5628</td>
<td>62.88</td>
</tr>
<tr>
<td>6048</td>
<td>63.11</td>
</tr>
<tr>
<td>6370</td>
<td>62.87</td>
</tr>
<tr>
<td>8041</td>
<td>63.10</td>
</tr>
</tbody>
</table>

Table A.52: Select results for carousel object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 2 part 1. See figure 4.93b
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>576</td>
<td>44.76</td>
</tr>
<tr>
<td>1000</td>
<td>33.64</td>
</tr>
<tr>
<td>1256</td>
<td>34.18</td>
</tr>
<tr>
<td>1446</td>
<td>34.42</td>
</tr>
<tr>
<td>1500</td>
<td>40.73</td>
</tr>
<tr>
<td>1547</td>
<td>38.73</td>
</tr>
<tr>
<td>4945</td>
<td>44.38</td>
</tr>
<tr>
<td>5015</td>
<td>39.55</td>
</tr>
<tr>
<td>7973</td>
<td>39.73</td>
</tr>
</tbody>
</table>

Table A.53: Select results for carousel object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 2 part 2. See figure 4.93c.

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>455</td>
<td>32.86</td>
</tr>
<tr>
<td>536</td>
<td>33.47</td>
</tr>
<tr>
<td>1000</td>
<td>19.18</td>
</tr>
<tr>
<td>1161</td>
<td>34.80</td>
</tr>
<tr>
<td>1256</td>
<td>34.93</td>
</tr>
<tr>
<td>1446</td>
<td>35.15</td>
</tr>
<tr>
<td>1500</td>
<td>35.47</td>
</tr>
<tr>
<td>5015</td>
<td>35.92</td>
</tr>
<tr>
<td>6960</td>
<td>36.08</td>
</tr>
</tbody>
</table>

Table A.54: Select results for carousel object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 3 part 1. See figure 4.93d.
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>480</td>
<td>19.14</td>
</tr>
<tr>
<td>637</td>
<td>14.31</td>
</tr>
<tr>
<td>938</td>
<td>11.53</td>
</tr>
<tr>
<td>1000</td>
<td>11.45</td>
</tr>
<tr>
<td>1256</td>
<td>11.61</td>
</tr>
<tr>
<td>1446</td>
<td>11.66</td>
</tr>
<tr>
<td>1500</td>
<td>11.76</td>
</tr>
<tr>
<td>4859</td>
<td>10.38</td>
</tr>
<tr>
<td>5015</td>
<td>13.43</td>
</tr>
</tbody>
</table>

Table A.55: Select results for carousel object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 3 part 2. See figure 4.93e
<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of Maximum MSC SPM Packet size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td>1000</td>
<td>59.77</td>
</tr>
<tr>
<td>1256</td>
<td>61.12</td>
</tr>
<tr>
<td>1446</td>
<td>61.79</td>
</tr>
<tr>
<td>1500</td>
<td>61.74</td>
</tr>
<tr>
<td>3819</td>
<td>62.60</td>
</tr>
<tr>
<td>5015</td>
<td>62.99</td>
</tr>
<tr>
<td>6370</td>
<td>63.00</td>
</tr>
<tr>
<td>7968</td>
<td>63.96</td>
</tr>
<tr>
<td>8041</td>
<td>63.63</td>
</tr>
</tbody>
</table>

Table A.56: Select results for carousel object over IP/MSC Datagroups/MSC EPM/Cambridge All Zeros 4. See figure 4.93f

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td>200</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>60.93</td>
</tr>
<tr>
<td>1256</td>
<td>60.93</td>
</tr>
<tr>
<td>1300</td>
<td><strong>63.83</strong></td>
</tr>
<tr>
<td>1446</td>
<td>63.83</td>
</tr>
<tr>
<td>1500</td>
<td>63.83</td>
</tr>
<tr>
<td>5015</td>
<td>63.83</td>
</tr>
</tbody>
</table>

Table A.57: Select results for carousel object over IP/MSC-FEC/Cambridge All Zeros 2 part 1. See figure 4.95a.
### Table A.58: Select results for carousel object over IP/MSC-FEC/Cambridge All Zeros 2 part 2. See figure 4.95b

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
<th>256</th>
<th>512</th>
<th>768</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>41.70</td>
<td>41.70</td>
<td>47.65</td>
<td>41.27</td>
<td>51.74</td>
</tr>
<tr>
<td>1000</td>
<td>41.70</td>
<td>41.70</td>
<td>47.65</td>
<td>41.27</td>
<td>51.74</td>
</tr>
<tr>
<td>1256</td>
<td>41.70</td>
<td>47.65</td>
<td>41.27</td>
<td>51.74</td>
<td></td>
</tr>
<tr>
<td>1300</td>
<td>41.70</td>
<td>47.65</td>
<td>41.27</td>
<td>51.74</td>
<td></td>
</tr>
<tr>
<td>1446</td>
<td>45.93</td>
<td>47.42</td>
<td>45.90</td>
<td>51.74</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>45.93</td>
<td>47.42</td>
<td>45.90</td>
<td>51.74</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>48.71</td>
<td>47.51</td>
<td>45.90</td>
<td>51.74</td>
<td></td>
</tr>
<tr>
<td>5015</td>
<td>48.71</td>
<td>47.51</td>
<td>45.90</td>
<td>51.74</td>
<td></td>
</tr>
</tbody>
</table>

### Table A.59: Select results for carousel object over IP/MSC-FEC/Cambridge All Zeros 3 part 1. See figure 4.95c

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
<th>256</th>
<th>512</th>
<th>768</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>51.06</td>
<td>58.03</td>
<td>51.06</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>51.06</td>
<td>58.03</td>
<td>51.06</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>58.03</td>
<td>58.03</td>
<td>51.06</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>1256</td>
<td>58.03</td>
<td>58.03</td>
<td>51.06</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>1300</td>
<td>60.79</td>
<td>58.03</td>
<td>60.79</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>1446</td>
<td>60.79</td>
<td>58.03</td>
<td>60.79</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>60.79</td>
<td>58.03</td>
<td>60.79</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>5015</td>
<td>60.79</td>
<td>58.03</td>
<td>60.79</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>Payload (bytes)</td>
<td>Performance of MPE-FEC row size (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>512</td>
<td>768</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td><strong>23.94</strong></td>
<td>23.94</td>
<td>23.94</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td><strong>27.11</strong></td>
<td>17.96</td>
<td><strong>24.87</strong></td>
<td>28.37</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>9.34</td>
<td>16.70</td>
<td>16.46</td>
<td><strong>30.22</strong></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>9.46</td>
<td>16.77</td>
<td>16.46</td>
<td><strong>30.22</strong></td>
<td></td>
</tr>
<tr>
<td>1256</td>
<td>9.46</td>
<td>16.77</td>
<td>16.46</td>
<td><strong>30.22</strong></td>
<td></td>
</tr>
<tr>
<td>1446</td>
<td>9.12</td>
<td>16.72</td>
<td>9.12</td>
<td><strong>30.22</strong></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>9.12</td>
<td>16.72</td>
<td>9.12</td>
<td><strong>30.22</strong></td>
<td></td>
</tr>
<tr>
<td>5015</td>
<td>16.82</td>
<td>16.77</td>
<td>9.17</td>
<td><strong>30.22</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table A.60: Select results for carousel object over IP/MSC-FEC/Cambridge All Zeros 3 part 2. See figure 4.95d

<table>
<thead>
<tr>
<th>Payload (bytes)</th>
<th>Performance of MPE-FEC row size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td>500</td>
<td>63.93</td>
</tr>
<tr>
<td>1000</td>
<td>65.83</td>
</tr>
<tr>
<td>1256</td>
<td>64.85</td>
</tr>
<tr>
<td>1300</td>
<td>67.72</td>
</tr>
<tr>
<td>1446</td>
<td>67.72</td>
</tr>
<tr>
<td>1500</td>
<td>67.46</td>
</tr>
<tr>
<td>3200</td>
<td>67.35</td>
</tr>
<tr>
<td>5015</td>
<td>67.35</td>
</tr>
</tbody>
</table>

Table A.61: Select results for carousel object over IP/MSC-FEC/Cambridge All Zeros 4. See figure 4.95e
Appendix B

Design for Software Simulator

B.1 Database Design

Figure B.1: Entity Relationship diagram of the DRBE database
List of Acronyms

1G—First Generation. (3, 9)

2G—Second Generation. (2, 9, 10)

2.5G—Enhanced Second Generation. (2, 9, 10)

3G—Third Generation. (2, 9)

AAC+—High Efficiency - Advanced Audio Coding. (12–14)

ARPANET—Advanced Research Projects Agency Network. (6)

BBC—British Broadcasting Corporation. (7, 13)

BCH—Bose-Chaudhuri-Hocquenghem. (18, 19)


BWS—Broadcast Website. (24, 33, 39)

CA—Conditional Access. (27, 33)

CDMA—Code Division Multiple Access. (10, 11)

CIF—Common Interleaved Frame. (25)

CRC—Cyclic Redundancy Check. (18, 19, 26, 27, 31–33, 39, 40, 43, 44, 60, 155, 204)

CU—Capacity Unit. (25)

DAB—Digital Audio Broadcast. (i, 1, 4, 5, 12–14, 18, 19, 22, 25, 26, 28, 31, 33, 34, 39–43, 45, 48–50, 52, 58, 61, 68, 77, 78, 80, 84, 85, 91, 96, 97, 101, 105, 106, 114, 115, 120, 121, 125, 135, 136, 140, 151, 154, 172, 198, 201–203, 205, 206)
DAB+—DAB with AAC+. (12–14, 25, 84)

DAB-IP—Internet Protocol over Digital Audio Broadcast. (116)

DAMPS—Digital Advanced Mobile Phone System. (9, 10)

DMB—Digital Multimedia Broadcast. (1, 4, 5, 13–15, 25, 26, 28, 39, 41, 42, 48–50, 80, 84, 114, 120, 121, 129, 165, 198, 201, 206)

DSC—Digital Selective Calling. (11)

DVB—Digital Video Broadcast. (12–14, 41, 43, 45, 125, 129, 140, 210)

DVB-c—Digital Video Broadcast - Cable. (13, 43)

DVB-h—Digital Video Broadcast - Hand-held. (i, 1, 4, 5, 13, 14, 19, 25, 43, 45, 48–50, 64–66, 80, 84, 125, 126, 129, 132, 135, 140, 141, 143, 148, 172, 190, 198, 201, 205, 206, 210)

DVB-s—Digital Video Broadcast - Satellite. (13, 43)

DVB-t—Digital Video Broadcast - Terrestrial. (12, 13, 43)

EBU—European Broadcasting Union. (12)

EDGE—Enhanced Data GSM Environment. (9, 11)


EXT—Extension Flag. (34, 36)


FIC—Fast Information Channel. (25)

FLUTE—File Delivery over Unidirectional Transport. (45, 46, 206)

FO—First Order. (46, 47)

GHz—Giga-Hertz. (3)

GMDSS—Global Maritime Distress and Safety System. (11)
GMSK—Gaussian minimum-shift keying. (11)

GPRS—General Packet Radio Service. (9–11)

GPS—Global Positioning System. (77)

GSM—Global System for Mobile Communications. (9–11)

IEEE—Institute of Electrical and Electronics Engineers. (8)

IMT-2000—International Mobile Telecommunications-2000. (9)


IR—Initialisation and Refresh. (46, 47)

IrDA—Infrared Data Association. (8)

ISO—International Organisation for Standardisation. (43)

ITU—International Telecommunication Union. (9)

kbps—kilobits per second. (9–11, 26, 43, 78, 160)

LAN—Local Area Network. (3, 8)

MAC—Media Access Control. (11)

MAN—Metropolitan Area Network. (3, 8)

mbps—megabits per second. (10)

MCI—Multiplex Configuration Information. (25)

modem—modulator/demodulator. (3, 9)

MOT—Multimedia Object Transfer. (15, 31, 33, 34, 36, 38, 39, 45)


MPEG—Moving Picture Experts Group. (14, 15)


MTU—Maximum Transmission Unit. (45, 114, 203)

MUX—MultipleX. (41, 42)

NAVTEX—NAVigational TEXT. (11)

NRSC—National Radio Systems Committee. (12)

PAN—Personal Area Network. (8)

PC—Personal Computer. (8)

PDA—Personal Digital Assistant. (8, 43)


PLI—Parameter Length Indicator. (34, 36)

QoS—Quality of Service. (80)

RBDS—Radio Broadcast Data System. (12)

RDS—Radio Data System. (12)

ROHC—RObust Header Compression. (24, 46, 47)


RTP—Real-time Transmission Protocol. (14, 24, 40, 45, 68, 116, 117, 119, 120, 126, 127)

RTP/UDP/IP—Real-time Transmission Protocol over User Datagram Protocol over Internet (24)

RTT—Round-Trip Time. (46)

SMS—Short Messaging Service. (9)
SO—Second Order. (46, 47)


TCP—Transmission Control Protocol. (6, 7, 46, 50)

TCP/IP—Transmission Control Protocol over Internet Protocol. (23)

TDC—Transparent Data Channel. (28, 33, 52–55, 57, 58, 64, 85, 86, 89, 91, 94, 96, 107, 110, 112, 119)

TDMA—Time Division Multiple Access. (9, 10)


UDP/IP—User Datagram Protocol over Internet Protocol. (7, 23)

UMTS—Universal Mobile Telecommunication System. (9, 10)

VHF—Very High Frequency. (11)

VJ Compression—Van Jacobson Compression. (23)

WAN—Wide Area Network. (6, 8)

WCDMA—Wideband Code Division Multiple Access. (10, 11)

WiFi—Wireless Fidelity. (3, 4, 8)

WiMAX—Worldwide Interoperability for Microwave Access. (3, 4, 8)